

32-bit Arm Cortex-M23 Low Power MCU with Security, Safety, CAN-FD, Full Speed USB, Touch and Advanced Analog

PIC32CM SG00/GC00



www.microchip.com
Product Page Links

32-bit Arm® Cortex®-M23 MCU with Security, Functional Safety (FuSa), CAN-FD, Full-Speed USB, Enhanced Touch, and Advanced Analog

Operating Conditions

- 1.71V to 3.63V, -40°C to +125°C, DC to 48 MHz
- 1.71V to 3.63V, -40°C to +85°C, DC to 72 MHz

Core: Arm Cortex-M23 CPU running at up to 72 MHz

- 2.64 CoreMark/MHz and 1.03 DMIPS/MHz
- Nested Vector Interrupt Controller (NVIC)
- Stack Limit Checking
- Memory Protection Unit (MPU)

Memories

- Additional 16 KB of Boot Flash Memory (BFM)
- Additional 32 KB of Configuration Flash Memory (CFM)
- Flash Supports:
 - Error Correction Code (ECC) with fault injection capability
 - CRC of any contiguous section
 - Deep Power Down option while system is in Standby
 - In-band error reporting for both read and write accesses
 - Tamper event logging
- 128 KB SRAM Main Memory with ECC and fault injection capability
 - Retained in Idle, Standby and Hibernate modes
- 512 Bytes TrustRAM

Security Features

- Arm TrustZone® technology for flexible hardware isolation of memories and peripherals
 - Configurable partitioning of PFM, BFM, SRAM
 - Individual security attribution for each peripheral, I/O, and external interrupt line
- Secure Boot (optional)
- Device Identity Composition Engine (DICE) support
- Physical Unclonable Function (PUF) generating a device unique key for local encryption and attestation

Security Features (Continued...)

- Hardware Security Module (HSM Lite)
 - AES (256 bits), SHA-1, SHA-2, RSA, ECC accelerator
 - True Random Number Generator
- TrustRAM

Advanced Analog Features and Touch

- 12-bit ADC module:
 - Up to 4.5 Msps
 - Up to 12 single-ended channels or 3 differential + 6 single-ended channels
 - External reference support
- Two Analog Comparators with programmable voltage references
- Enhanced Peripheral Touch controller (PTC)
 - Up to 324 (18 x 18) mutual capacitance channels
 - Up to 36 self-capacitance channels with Driven Shield+ technology for better noise immunity and moisture tolerance
 - Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels
 - Hardware noise filtering and noise signal desynchronization for high conducted immunity
 - Supports wake-up on touch from Standby Sleep mode
 - Supports large self-capacitor sensor

System

- Integrated Power-on Reset (POR) and programmable Brown-out Detection (BOD) on VDDIO and VDDREG
- Programmable Low-Voltage Detect Module (LVD)
- 12-channel event system for Inter-peripheral Core-independent Operation (EVSYS)
- Unique 128-bit serial number

Hardware Safety Features

- ECC w/ Fault Injection capability on Flash and SRAM
- SRAM/TrustRAM MBIST on boot User Mode accessible
- IP registers write protection through PAC
- Fail-Safe Clock Monitor (CFD)
- ISO 26262:2018 Compliant

Power Management

- Idle mode for fast wake up time
- Standby mode, Backup mode, Off mode, and SleepWalking Peripherals
- Hibernate mode up to full SRAM retention

Timers/Output Compare/Input Capture

- Up to seven 16-bit Timer/Counters for Control (TCC), each with two double buffered compare/capture channels
- 32-bit Real-Time Counter (RTC) with clock/calendar functions
- Watchdog Timer (WDT) with Window mode

Clock Management

- 4 MHz to 48 MHz Crystal Oscillator (XOSC48)
 - Clock failure detection with safe clock switch
- 32.768 kHz crystal oscillator (XOSC32K)
 - Clock failure detection with safe clock switch
- 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)
- 48 MHz digital Phase-Locked Loop (DFLL48M)
- 1.6 GHz Phase Locked Loop (PLL1G6)
- Frequency meter (FREQM)

Software and Tools Support: Develop Prototypes Quickly with A Powerful, Easy-to-Use Ecosystem

- Get code off to a head start with MPLAB Code Configurator
- Graphically configure peripherals, software libraries, and supported RTOS with MPLAB Harmony v3
- Download MPLAB XC Compiler
- Take advantage of MPLAB X IDE's support for 32-bit MCUs
- Select the best debugger for the project: MPLAB® ICE, MPLAB ICD, or PICKIT™

Direct Memory Access (DMA)

- Eight channels with 4 different block transfer modes
- Programmable 32-bit Cyclic Redundancy Check (CRC)

Input/Output

- High current pins with up to 25 mA source/sink
- Configurable Open-Drain Output on Digital I/O Pins
- Up to 10 5V-tolerant Input Pins (digital pins only)
- Up to 80 programmable I/O lines
- 16 external interrupts (EIC)
- Two Configurable Custom Logic (CCL) that supports:
 - Combinatorial logic functions, such as AND, NAND, OR, and NOR
 - Sequential logic functions, such as Flip-Flop and Latches
- One general purpose LDO output
 - 1.2/1.5/1.8/2.5V generated from VDDIO
 - Up to 100 mA
- Two outputs controlled by SUPC

Communication Interfaces /Digital Peripherals

- Two CAN-FD modules (ISO 11898-1:2015)
 - Support CAN 2.0 A/B
- Six Serial Communication Interfaces (SERCOM), each configurable to operate as:
 - USART with full-duplex and single wire half-duplex configuration
 - I²C Host/Client up to 3.4 MHz
 - SPI
 - RS-485, IRDA, LIN Host/Client
- One Full-Speed (12 Mbps) Universal Serial Bus (USB) 2.0 interface
 - Embedded Host and device function
 - Eight endpoints
 - On-chip transceiver with integrated serial resistor
 - Support crystal-less operation in device mode

Qualification

- AEC-Q100 Grade 1 (-40°C to +125°C)

Debugger Development Support

- In-circuit and in-application programming/debugging with SWD and JTAG
- Cortex-M debugger port
- Supports 8 breakpoints and 4 watch points
- IEEE®1149-compatible (JTAG) boundary scan
- Non-intrusive hardware-based instruction trace, Secure Debugging

Table 1. Packages

Type	VQFN		TQFP		
Pin Count	48	64	48	64	100
I/O Pins (up to)	36	50	36	50	80
Contact/Lead Pitch	0.5 mm	0.5 mm	0.5 mm	0.5 mm	0.5 mm
Dimensions	7x7x0.9 mm	9x9x1 mm	7x7x1 mm	10x10x1 mm	14x14x1 mm

Table of Contents

32-bit Arm® Cortex®-M23 MCU with Security, Functional Safety (FuSa), CAN-FD, Full-Speed USB, Enhanced Touch, and Advanced Analog.....	1
1. Guidelines for Getting Started.....	12
1.1. Basic Connection Requirements.....	12
1.2. Decoupling Capacitors.....	12
1.3. External Reset (RESET) Pin.....	13
1.4. Debugging or Programming Pins.....	13
1.5. JTAG.....	13
1.6. External Oscillator Pins.....	13
1.7. Unused I/Os.....	14
1.8. Considerations When Interfacing to Remotely Powered Circuits.....	14
1.9. Designing for High-Speed Peripherals.....	16
2. Configuration Summary.....	19
3. Ordering Information.....	20
4. Block Diagram.....	21
5. Pinout and Packaging.....	22
5.1. 48-pin TQFP and 48-pin VQFN.....	22
5.2. 64-pin TQFP and 64-pin VQFN.....	24
5.3. 100-pin TQFP.....	27
6. Signal Description.....	30
7. Power Supplies and Startup Considerations.....	40
7.1. Power Domain Overview.....	40
7.2. Power Domain Constraints.....	40
7.3. Power Up.....	42
7.4. Analog Peripherals Considerations.....	42
7.5. Device Startup.....	42
8. Product Mapping.....	44
9. Peripherals.....	47
9.1. Register Description.....	47
10. Processor and Architecture.....	67
10.1. Processor and Architecture.....	67
10.2. Arm Cortex-M23 Configuration.....	67
10.3. Nested Vector Interrupt Controller.....	67
10.4. NVIC Interrupt Mapping.....	67
10.5. High Speed Bus.....	71
10.6. Bus Matrix Connectivity.....	71
11. Memories.....	72
11.1. Embedded Memories.....	72
11.2. Physical Memory Map.....	72

11.3. SRAM Memory Configuration.....	72
11.4. Flash Memories.....	72
11.5. Unique ID (UID).....	74
12. PIC32CM SG00/GC00 Security Features.....	75
12.1. Data Flash.....	75
12.2. TrustRAM.....	75
12.3. Arm TrustZone for ARMv8-M.....	75
13. Implementation Defined Attribution Unit (IDAU).....	78
13.1. Overview.....	78
13.2. IDAU Regions IDS.....	78
13.3. IDAU Region Types.....	78
13.4. Secure and Secure Always On Region Types.....	79
13.5. IDAU Programmable Regions.....	80
13.6. IDAU Watermark Regions.....	81
13.7. H2PB Exempt Main Regions.....	87
13.8. IDAU Regions Configuration.....	87
13.9. Configuration Write Lock.....	89
13.10. Peripheral Dependencies.....	89
13.11. Register Summary.....	90
14. System Bus AHB - APB Bridge (H2PB).....	99
14.1. Overview.....	99
14.2. Features.....	99
14.3. Functional Description.....	99
14.4. Peripheral Dependencies.....	100
14.5. Register Summary.....	101
15. Multi-Channel RAM Controller (MCRAMC).....	108
15.1. Overview.....	108
15.2. Features.....	108
15.3. Block Diagram.....	108
15.4. Functional Description.....	108
15.5. Peripheral Dependencies.....	111
15.6. Register Summary.....	112
16. Peripheral Access Controller (PAC).....	124
16.1. Overview.....	124
16.2. Features.....	124
16.3. Debug Operation.....	124
16.4. Register Access Protection.....	124
16.5. Functional Description.....	124
16.6. Peripheral Dependencies.....	126
16.7. Register Summary.....	127
17. Device Service Unit (DSU).....	138
17.1. Overview.....	138
17.2. Features.....	138
17.3. DSU Block Diagram.....	139

17.4. Signal Description.....	139
17.5. Debug Operation.....	139
17.6. Programming.....	142
17.7. Device Identification.....	142
17.8. Peripheral Dependencies.....	143
17.9. Register Summary.....	144
18. Clock Distribution System.....	181
18.1. Clock Distribution.....	181
18.2. Clock System Features.....	181
18.3. Synchronous and Asynchronous Clocks.....	182
18.4. Register Synchronization.....	182
18.5. Enabling a Peripheral.....	184
18.6. On Demand Clock Requests.....	185
18.7. Power Consumption Versus Speed.....	185
18.8. Clocks after Reset.....	186
19. Oscillator Controller (OSCCTRL).....	187
19.1. Overview.....	187
19.2. Features.....	187
19.3. OSCCTRL Block Diagram.....	188
19.4. Signal Descriptions.....	188
19.5. Functional Description.....	188
19.6. Peripheral Dependencies.....	201
19.7. Register Summary.....	202
20. Generic Clock Controller (GCLK).....	248
20.1. Overview.....	248
20.2. Features.....	248
20.3. Block Diagram.....	248
20.4. Signal Description.....	249
20.5. Functional Description.....	249
20.6. Peripheral Dependencies.....	254
20.7. GCLK Peripheral Channel Control Mapping.....	254
20.8. Register Summary.....	256
21. Main Clock (MCLK).....	264
21.1. Overview.....	264
21.2. Features.....	264
21.3. Block Diagram.....	264
21.4. Functional Description.....	265
21.5. Peripheral Dependencies.....	266
21.6. MCLK AHB/APB Clock Enable Peripheral Mapping.....	267
21.7. Register Summary.....	269
22. 32 KHz Oscillators Controller (OSC32KCTRL).....	276
22.1. Overview.....	276
22.2. Features.....	276
22.3. Block Diagram.....	277
22.4. Signal Description.....	277

22.5. Functional Description.....	277
22.6. Peripheral Dependencies.....	281
22.7. Register Summary.....	282
23. Watchdog Timer (WDT).....	298
23.1. Overview.....	298
23.2. Features.....	298
23.3. Block Diagram.....	299
23.4. Functional Description.....	299
23.5. Peripheral Dependencies.....	303
23.6. Register Summary.....	304
24. Frequency Meter (FREQM).....	315
24.1. Overview.....	315
24.2. Features.....	315
24.3. Block Diagram.....	315
24.4. Signal Description.....	315
24.5. Clocks.....	316
24.6. Functional Description.....	316
24.7. Peripheral Dependencies.....	319
24.8. Register Summary.....	320
25. Real-Time Counter (RTC).....	335
25.1. Overview.....	335
25.2. Features.....	335
25.3. Block Diagram.....	336
25.4. Signal Description.....	337
25.5. Functional Description.....	337
25.6. Peripheral Dependencies.....	348
25.7. Register Summary - MODE0.....	349
25.8. Register Summary - MODE1.....	364
25.9. Register Summary - MODE2.....	380
25.10. Register Summary.....	395
26. Direct Memory Access Controller (DMAC).....	406
26.1. Overview.....	406
26.2. Features.....	406
26.3. Block Diagram.....	408
26.4. Functional Description.....	408
26.5. Peripheral Dependencies.....	428
26.6. Register Summary.....	429
26.7. DMA Descriptor Register Summary.....	458
27. Supply Controller (SUPC).....	465
27.1. Overview.....	465
27.2. Features.....	465
27.3. Block Diagram.....	466
27.4. Signals Description.....	468
27.5. Functional Description.....	468
27.6. Peripheral Dependencies.....	472

27.7. Register Summary.....	473
28. Power Manager (PM).....	495
28.1. Overview.....	495
28.2. Features.....	495
28.3. Block Diagram.....	495
28.4. Functional Description.....	495
28.5. Peripheral Dependencies.....	501
28.6. Register Summary.....	502
29. Reset Controller (RSTC).....	510
29.1. Overview.....	510
29.2. Features.....	510
29.3. Block Diagram.....	510
29.4. Signals Description.....	510
29.5. Functional Description.....	511
29.6. Peripheral Dependencies.....	512
29.7. Register Summary.....	513
30. External Interrupt Controller (EIC).....	518
30.1. Overview.....	518
30.2. Features.....	518
30.3. Block Diagram.....	518
30.4. Signal Description.....	519
30.5. Interrupts.....	519
30.6. Events.....	519
30.7. Sleep Mode Operation.....	519
30.8. Secure Access Right.....	519
30.9. Synchronization.....	520
30.10. Peripheral Dependencies.....	520
30.11. Register Summary.....	521
31. Non-Volatile Memory Controller (NVMCTRL).....	546
31.1. Flash Controller, Write.....	546
31.2. Flash Controller, Read.....	562
31.3. Peripheral Dependencies.....	580
31.4. Peripheral Dependencies.....	581
31.5. Register Summary.....	582
31.6. Register Summary.....	634
32. Event System (EVSYS).....	676
32.1. Overview.....	676
32.2. Features.....	676
32.3. Block Diagram.....	676
32.4. Power Management.....	677
32.5. Clocks.....	677
32.6. Functional Description.....	677
32.7. Peripheral Dependencies.....	683
32.8. Register Summary.....	684

33. I/O Pin Controller (PORT).....	708
33.1. Overview.....	708
33.2. Features.....	708
33.3. Block Diagram.....	709
33.4. Signal Description.....	709
33.5. I/O Lines.....	709
33.6. Clocks.....	709
33.7. Power Management.....	709
33.8. Debug Operation.....	710
33.9. Register Access Protection.....	710
33.10. Analog Connections.....	710
33.11. PIC32CM SG Specific Secure Access Protection.....	710
33.12. Functional Description.....	711
33.13. Peripheral Dependencies.....	716
33.14. Register Summary.....	717
34. Serial Communication Interface (SERCOM).....	748
34.1. Overview.....	748
34.2. Features.....	748
34.3. Block Diagram.....	748
34.4. Functional Description.....	749
34.5. Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART).....	753
34.6. SERCOM Serial Peripheral Interface (SPI).....	768
34.7. SERCOM I ² C.....	784
34.8. Peripheral Dependencies.....	808
34.9. Register Summary - USART.....	810
34.10. Register Summary - SPI.....	839
34.11. Register Summary - I2CM.....	864
34.12. Register Summary - I2CS.....	889
35. Controller Area Network (CAN).....	913
35.1. Overview.....	913
35.2. Features.....	913
35.3. Block Diagram.....	914
35.4. Signal Description.....	914
35.5. Functional Description.....	914
35.6. Peripheral Dependencies.....	944
35.7. Register Summary.....	945
35.8. Rx Buffer and FIFO Element Register Summary.....	1058
35.9. Tx Buffer Element Register Summary.....	1061
35.10. Tx Event FIFO Element Register Summary.....	1064
35.11. Standard Message ID Filter Element Register Summary.....	1066
35.12. Extended Message ID Filter Element Register Summary.....	1068
36. Universal Serial Bus (USB).....	1071
36.1. Overview.....	1071
36.2. Features.....	1071
36.3. USB Block Diagram.....	1072
36.4. Signal Description.....	1072

36.5. Functional Description.....	1072
36.6. Peripheral Dependencies.....	1089
36.7. Register Summary.....	1090
36.8. Register Summary - DEVICE.....	1112
36.9. Register Summary - HOST.....	1125
36.10. USB Descriptor Register Summary.....	1135
37. Configurable Custom Logic (CCL).....	1146
37.1. Overview.....	1146
37.2. Features.....	1146
37.3. Block Diagram.....	1147
37.4. Signal Description.....	1147
37.5. I/O Lines.....	1147
37.6. Power Management	1147
37.7. Clocks.....	1147
37.8. Events.....	1148
37.9. Debug Operation.....	1148
37.10. Register Access Protection.....	1148
37.11. Functional Description.....	1148
37.12. Events.....	1156
37.13. Sleep Mode Operation.....	1156
37.14. Peripheral Dependencies.....	1156
37.15. Register Summary.....	1158
38. Analog to Digital Converter (ADC).....	1164
38.1. Overview.....	1164
38.2. Features.....	1164
38.3. Block Diagram.....	1165
38.4. Signal Description.....	1168
38.5. Functional Description.....	1168
38.6. Peripheral Dependencies.....	1180
38.7. Register Summary.....	1181
39. Analog Comparator (AC).....	1243
39.1. Overview.....	1243
39.2. Features.....	1243
39.3. Block Diagram.....	1244
39.4. Analog Connections.....	1244
39.5. Functional Description.....	1245
39.6. Peripheral Dependencies.....	1254
39.7. Register Summary.....	1255
40. Timer/Counter for Control Applications (TCC).....	1277
40.1. Overview.....	1277
40.2. Features.....	1277
40.3. Block Diagram.....	1278
40.4. Signal Description.....	1279
40.5. Functional Description.....	1280
40.6. Peripheral Dependencies.....	1324
40.7. Register Summary.....	1328

41. TrustRAM (TRAM).....	1369
41.1. Overview.....	1369
41.2. Features.....	1369
41.3. Block Diagram.....	1369
41.4. Functional Description.....	1369
41.5. Peripheral Dependencies.....	1373
41.6. Register Summary.....	1374
42. Peripheral Touch Controller (PTC).....	1388
42.1. Overview.....	1388
42.2. Features.....	1388
42.3. Block Diagram.....	1389
42.4. Signal Description.....	1389
42.5. Peripheral Dependencies.....	1390
42.6. Functional Description.....	1391
42.7. Peripheral Dependencies.....	1393
43. Hardware Security Module Lite (HSM-Lite).....	1394
43.1. Overview.....	1394
43.2. Features.....	1394
43.3. Operation.....	1394
43.4. Cryptographic Accelerators.....	1395
43.5. TRNG.....	1395
43.6. PUF Interface.....	1395
43.7. Peripheral Dependencies.....	1395
43.8. Register Summary.....	1396
44. Anti-Tamper Module (AT).....	1465
44.1. Overview.....	1465
45. Physically Uncloneable Function (PUF).....	1466
45.1. Overview.....	1466
46. Electrical Characteristics.....	1467
46.1. Absolute Maximum Electrical Characteristics.....	1467
46.2. CPU Electrical Characteristics.....	1468
46.3. Power Supply.....	1468
46.4. Active Power.....	1471
46.5. Idle Power.....	1472
46.6. Standby Power.....	1474
46.7. Hibernate Power.....	1476
46.8. Backup Power.....	1478
46.9. OFF Power.....	1478
46.10. Wake-Up Timing.....	1479
46.11. Peripheral Active Power.....	1479
46.12. I/O Pin Electrical Characteristics.....	1480
46.13. Internal Voltage Reference Electrical Specifications (IVREF).....	1484
46.14. Maximum Clock Frequencies.....	1485
46.15. XOSC Electrical Characteristics.....	1486
46.16. XOSC32K Electrical Characteristics.....	1489

46.17. OSCULP32K Electrical Characteristics.....	1492
46.18. DFLL-FDPLL Electrical Characteristics.....	1492
46.19. Analog-to-Digital Converter (ADC) Electrical Specifications.....	1494
46.20. Comparator (AC) Electrical Characteristics.....	1497
46.21. Peripheral Touch Controller (PTC) Electrical Characteristics.....	1499
46.22. Serial Peripheral Interface (SPI) Timings Electrical Characteristics.....	1500
46.23. Universal Asynchronous Receiver/Transmitter (UART) Electrical Characteristics.....	1506
46.24. I ² C Electrical Characteristics.....	1507
46.25. Controller Area Network (CAN) Electrical Characteristics.....	1514
46.26. Timer/Counter Controller (TCC) Electrical Characteristics.....	1514
46.27. Universal Serial Bus (USB) Electrical Characteristics.....	1516
46.28. Flash Controller Electrical Characteristics.....	1516
46.29. Frequency Meter (FREQM) Electrical Characteristics.....	1517
46.30. JTAG Electrical Characteristics.....	1518
46.31. SWD 2-Wire Electrical Characteristics.....	1519
46.32. Configurable Custom Logic (CCL) Electrical Specifications.....	1519
46.33. User LDO Electrical Specifications.....	1520
46.34. Multi-Channel Ram Controller (MCRAMC) Electrical Specifications.....	1520
47. Packaging Information.....	1521
47.1. Package Marking Information.....	1521
47.2. Package Drawings.....	1521
47.3. Soldering Profile.....	1536
48. Schematic Checklist.....	1538
48.1. Introduction.....	1538
49. Common Conventions.....	1539
49.1. Numerical Notation.....	1539
49.2. Memory Size and Type.....	1539
49.3. Frequency and Time.....	1539
49.4. Registers and Bits.....	1539
50. Acronyms and Abbreviations.....	1541
51. Revision History.....	1543
Microchip Information.....	1546
Trademarks.....	1546
Legal Notice.....	1546
Microchip Devices Code Protection Feature.....	1546
Product Page Links.....	1547

1. Guidelines for Getting Started

1.1 Basic Connection Requirements

Getting started with the PIC32CM SG00/GC00 Family of 32-bit Microcontrollers (MCUs) requires attention to a minimal set of device pin connections before proceeding with development. The following is a list of pin names, which must be connected always:

- All VDDIO and VSS pins (see [Decoupling Capacitors](#))
- All AVDD and AVSS pins, even if the ADC module is not used (see [Decoupling Capacitors](#))
- The RESET pin (see [External RESET Pin](#))
- The SWDIO, SWCLK pins, used for In-Circuit Programming and debugging purposes (see [Debug/Programming Pins](#))
- The XIN/XOUT and XIN32/XOUT32 pins, when external oscillator source is used (see [External Oscillator Pins](#))

The following pins may be required:

- The ADC_VREFH pin, used when external voltage reference for the ADC module is implemented
- TDI, TMS, TDO and TCK if JTAG is used

1.2 Decoupling Capacitors

The use of decoupling capacitors on power supply pins, such as VREG, VDDIO, and AVDD is required, see the [Schematic Checklist](#) chapter.

Consider the following criteria when using decoupling capacitors:

- **Bulk capacitors:** Must be utilized on all power pins as indicated in the [Schematic Checklist](#) chapter.
- **Value and type of capacitor:** A value of 0.1 μF (100 nF), 10-20V is recommended. The capacitor should be a low Equivalent Series Resistance (low-ESR) capacitor and have resonance frequency in the range of 20 MHz and higher. It is further recommended that ceramic capacitors be used.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended that the capacitors be placed on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is within one-quarter inch (6 mm) in length.
- **Handling high frequency noise:** If the board is experiencing high frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF . Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible. For example, 0.1 μF in parallel with 0.001 μF .
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum thereby reducing PCB track inductance.

1.2.1 Bulk Capacitors

The use of a bulk capacitor is recommended to improve power supply stability. Typical values range from 4.7 μF to 22 μF ceramic or tantalum capacitors with low ESR. This capacitor should be located as close to the device as possible, refer to the [Schematic Checklist](#) chapter.

1.3 External Reset ($\overline{\text{RESET}}$) Pin

Pulling the $\overline{\text{RESET}}$ pin low generates a device Reset.

For example, as illustrated in the [Schematic Checklist](#) chapter, it is recommended that the capacitor C, be isolated from the $\overline{\text{RESET}}$ pin. Place the components illustrated within one-half inch (12 mm) from the $\overline{\text{RESET}}$ pin.

1.4 Debugging or Programming Pins

The SWDIO and SWCLK pins are used for In-Circuit programming and debugging purposes. It is recommended to keep the trace length between the debug external connector and the debug pins on the device as short as possible to minimize ESD/EMI vulnerabilities. If the debug external connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ω s, not to exceed 100 Ω s with protection using Transient Voltage Suppressors (TVS), at the user's discretion. Refer to the [Schematic Checklist](#) chapter.

1.5 JTAG

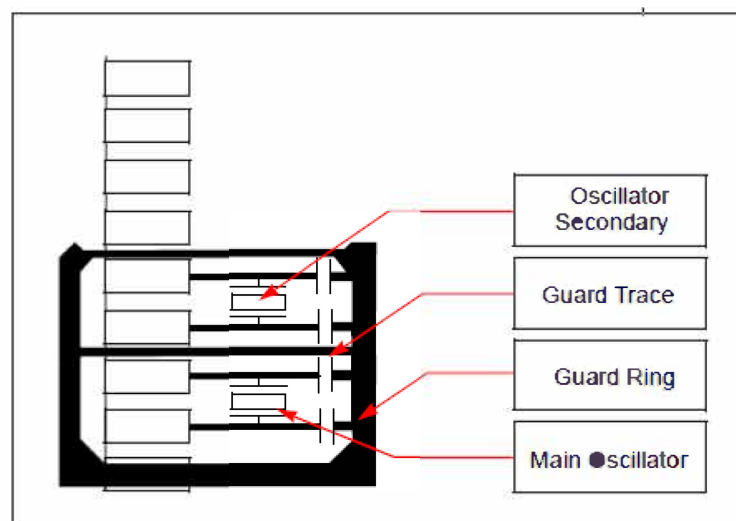
The TMS, TDO, TDI and TCK pins are used for testing, debugging, and boundary scan according to the Joint Test Action Group (JTAG) standard. It is recommended to keep the trace length between the JTAG connector and the JTAG pins on the device as short as possible. If the JTAG connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ω s, not to exceed 100 Ω s with protection using Transient Voltage Suppressors (TVS), at the user's discretion. Refer to the [Schematic Checklist](#) chapter.

1.6 External Oscillator Pins

Many MCUs have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator.

The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate them from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. A suggested layout is illustrated in the following figure.

Figure 1-1. Suggested Oscillator Circuit Placement



1.6.1 Crystal Oscillator Design Consideration

The following hypothetical example assumptions are used to calculate the Primary Oscillator loading capacitor values:

- C_{IN} = XIN Pin Capacitance = 4 pF
- C_{OUT} = XOUT Pin Capacitance = 4 pF
- PCB stray capacitance (i.e., 12 mm length) = 2.5 pF
- C_1 and C_2 = The loading capacitors to use on your crystal circuit design to guarantee that the effective capacitance as seen by the crystal in circuit meets the crystal manufacturer specification MFG Crystal Data Sheet CLOAD specification:

$$C_{LOAD} = \{ ([C_{IN} + C_1] * [C_{OUT} + C_2]) / [C_{IN} + C_1 + C_2 + C_{OUT}] \} + \text{oscillator PCB stray capacitance}$$

Hypothetical Example Crystal Load Capacitor Calculation

Crystal manufacturer data sheet spec example: $C_{LOAD} = 15 \text{ pF}$

Therefore,

$$MFG \ C_{LOAD} = \{ ([C_{IN} + C_1] * [C_{OUT} + C_2]) / [C_{IN} + C_1 + C_2 + C_{OUT}] \} + \text{estimated oscillator PCB stray capacitance.}$$

Assuming $C_1 = C_2$ and PIC32C $C_{in} = C_{out}$, the formula can be further simplified and restated to solve for C_1 and C_2 by: $C_1 = C_2 = ((2 * MFG \ C_{Load \ spec}) - C_{in} - (2 * PCB \ capacitance)) / ((2 * 15) - 4 - (2 * 2.5 \text{ pF})) = (30 - 4 - 5) = 21 \text{ pF}$

Therefore,

$C_1 = C_2 = 21 \text{ pF}$ is the correct loading capacitors to use on your crystal circuit design to guarantee that the effective capacitance as seen by the crystal in circuit in this example is 15 pF to meet the crystal manufacturer specification.

Note: Do not add excessive gain such that the oscillator signal is clipped, flat on top of the sine wave. If so, you need to reduce the gain accordingly. Failure to do so can stress and age the crystal, which can result in an early failure. When measuring the oscillator signal you must use an FET active-powered scope probe with 1 pF or the scope probe itself will unduly change the gain and peak-to-peak levels.

1.7 Unused I/Os

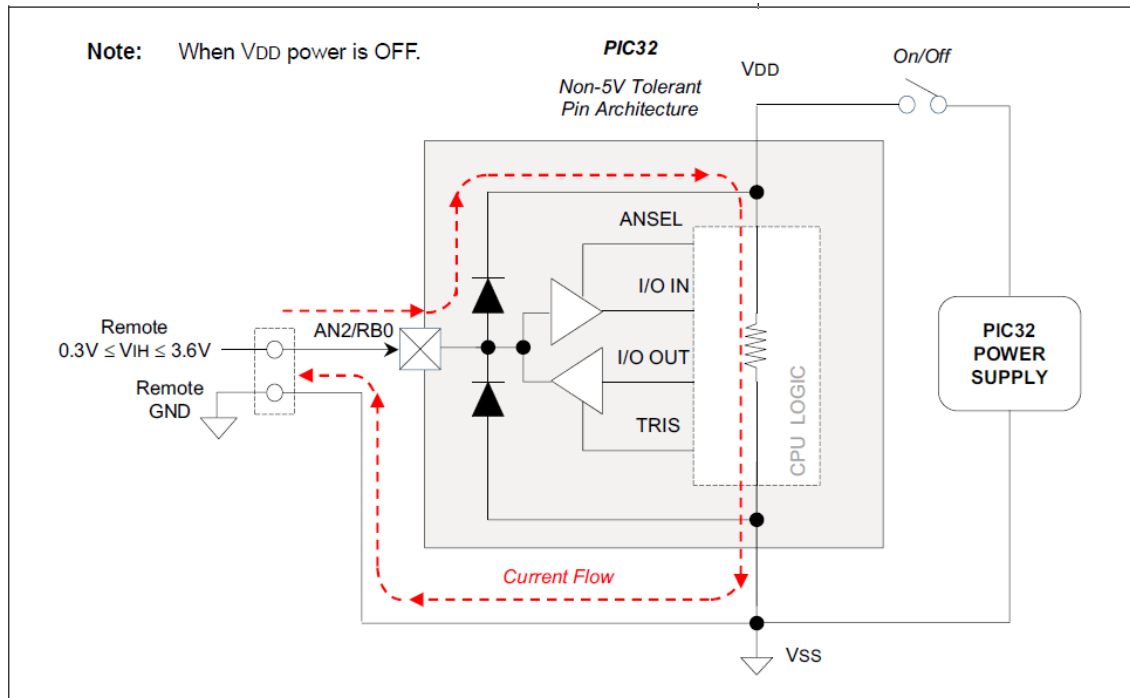
Unused I/O pins should not be allowed to float as inputs. It is recommended that unused inputs be ganged together and connected through a 1K resistor to digital ground or individually or in multiple groups through 1K as the PCB layout permits. This minimizes the chip vulnerability to ESD and radiated EMI due to a high voltage discharge event.

1.8 Considerations When Interfacing to Remotely Powered Circuits

1.8.1 Non-5V Tolerant Input Pins

A quick review of the absolute maximum rating in the [Electrical Characteristics](#) will indicate that the voltage on any non-5V tolerant pin may not exceed $V_{DD} + 0.3V$ unless the input current is limited to meet the respective injection current specifications defined by electrical spec parameters DI_19, DI_21, and DI_23 as described in *I/O Pin AC/DC Electrical Specifications*. The following figure shows an example of a remote circuit using an independent power source, which is powered while connected to a PIC32C non-5V tolerant circuit that is not powered.

Figure 1-2. PIC32C Non-5V Tolerant I/O Pin Circuit Example



Without proper signal isolation, on non-5V tolerant pins, the remote signal can power the PIC32C device through the high side ESD protection diodes. Besides violating the absolute maximum rating specification when VDD of the PIC32C device is restored and ramping up or ramping down, it can also negatively affect the internal Power-on Reset (POR) and Brown-out Reset (BOR) circuits, which can lead to improper initialization of internal PIC32C logic circuits. In these cases, it is recommended to implement digital or analog signal isolation as depicted in the following figure, as appropriate or use series I/O pin resistors when possible to limit the injection current to less than 5 mA/pin and/or 20 mA total for the whole MCU. This is indicative of all industry microcontrollers and not just Microchip products.

Figure 1-3. Example Digital/Analog Signal Isolation Circuits

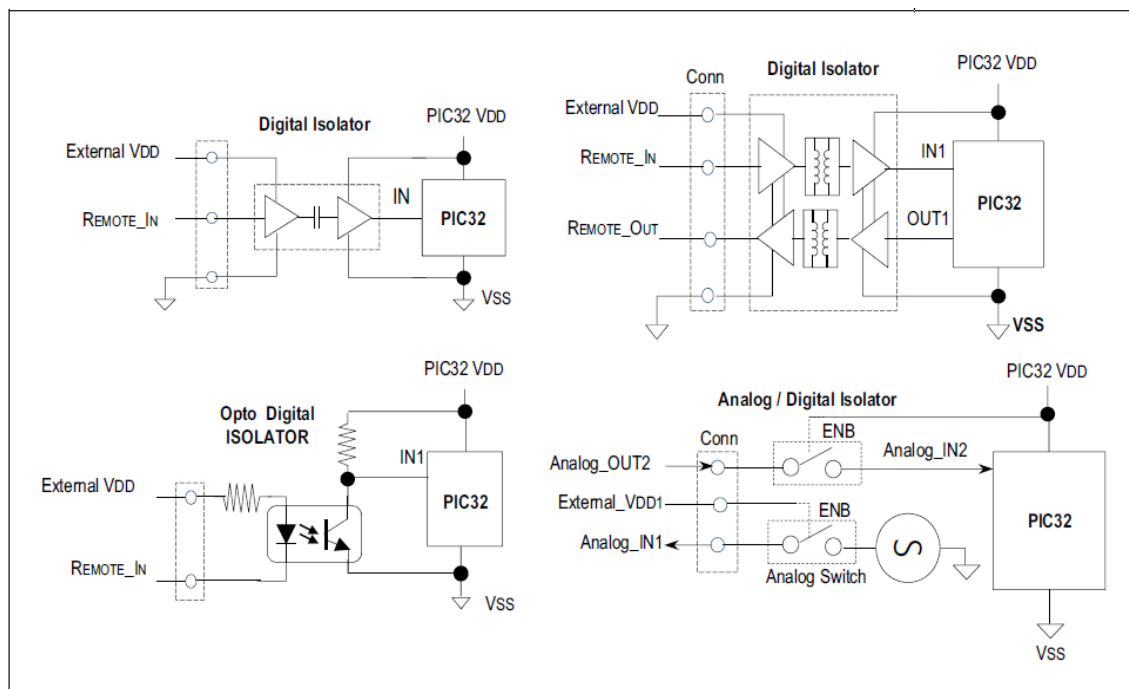


Table 1-1. Examples of Digital/Analog Isolators with Optional Level Translation

Example Digital/ Analog Signal Isolation Circuits	Inductive Coupling	Capacitive Coupling	Opto Coupling	Analog/Digital Switch
ADuM7241 / 40 ARZ (1 Mbps)	X	—	—	—
ADuM7241 / 40 CRZ (25 Mbps)	X	—	—	—
ISO721	—	X	—	—
LTV-829S (2 Channel)	—	—	X	—
LTV-849S (4 Channel)	—	—	X	—
FSA266 / NC7WB66	—	—	—	X

1.9 Designing for High-Speed Peripherals

The PIC32C Family of devices have peripherals that operate at frequencies much higher than typical for an embedded environment. The following list shows the peripherals that produce high-speed signals on their external pins on this device:

- USB
- SPI (Sercom)

Due to these high-speed peripheral signals, it is important to consider several factors when designing a product that uses these peripherals, as well as the PCB on which these components will be placed. Adhering to these recommendations will help achieve the following goals:

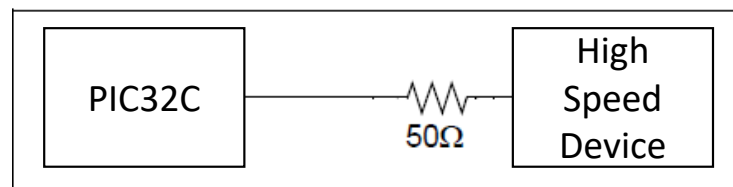
- Minimize the effects of electromagnetic interference for the proper operation of the product.
- Run all PCB high-speed signals first on component side of PCB.
- Ensure signals arrive at their intended destination at the same time by matching critical trace lengths on the PCB.
- Minimize crosstalk. Insure continuous ground under all high-speed signals.
- Maintain signal integrity by the use of termination resistors in the 30-50 Ω range.
- Reduce system noise by using bulk and high frequency decoupling caps and inductors on power rails.
- Minimize ground bounce and power sag. Use a dedicated ground plane if possible or at a minimum a star ground configuration. Do not daisy chain ground and power traces to components.

1.9.1 System Design

1.9.1.1 Impedance Matching

When selecting parts to place on high-speed signal bus, if the remote I/O pin impedance of the peripheral device does not match the impedance of the pins on the PIC32C device to which it is connected, signal reflections could result, thereby degrading the quality of the signal. If it is not possible to select a product that matches impedance, place a series resistor at the load to create the matching impedance. See the following figure for an example.

Figure 1-4. Series Resistor



1.9.1.2 PCB Layout Recommendations

The following recommendations will help ensure the PCB layout will promote the goals previously listed.

- **Component Placement:**
 - Place bypass capacitors as close to their component power and ground pins as possible, and place them on the same side of the PCB
 - Devices on the same bus that have larger setup times must be placed closer to the device
- **Power and Ground:**
 - Multi-layer PCBs will allow separate power and ground planes
 - Each ground pin should be connected to the ground plane individually
 - Place bypass capacitor vias as close to the pad as possible (preferably inside the pad)
 - If power and ground planes are not used, maximize width for power and ground traces
 - Use low-ESR, surface-mount bypass capacitors
- **Clocks and Oscillators:**
 - Place crystals as close as possible to the device XIN/XOUT and XIN32/XOUT32 pins
 - Do not route high-speed signals near the clock or oscillator
 - Avoid via usage and branches in high-speed clock lines

- Place termination resistors at the end of clock lines
- **Traces:**
 - Higher-priority signals must have the shortest traces
 - Avoid long run lengths on parallel traces to reduce coupling
 - Make the clock traces as straight as possible
 - Use rounded turns rather than right-angle turns
 - Have traces on different layers intersect on right angles to minimize crosstalk
 - Maximize the distance between traces, preferably no less than three times the trace width
 - Power traces should be as short and as wide as possible
 - High-speed traces must have a continuous ground beneath them

1.9.1.3 EMI/EMC/EFT (IEC 61000-4-4 and IEC 61000-4-2) Supression Considerations

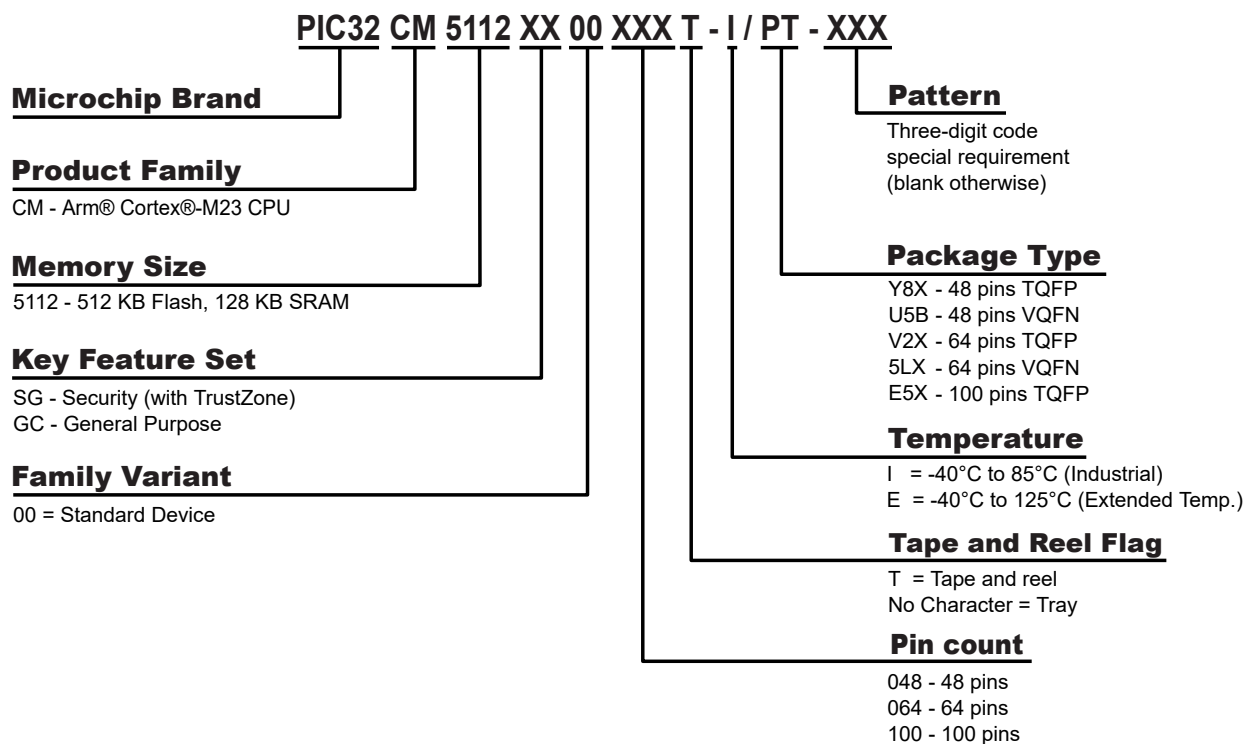
The use of LDO regulators is preferred to reduce overall system noise and provide a cleaner power source. However, when utilizing switching Buck/Boost regulators as the local power source for PIC32C devices, as well as in electrically noisy environments or test conditions required for IEC 61000-4-4 and IEC 61000-4-2, users should evaluate the use of Pie-Filters (i.e., L-C) on the power pins, as shown in the [Schematic Checklist](#) chapter. In addition to a less noisy power source, use of this type of T-Filter can greatly reduce susceptibility to EMI sources and events. Use Transient Voltage Suppressors (TVS) on power buses as well as on all external PCB signal connections. If design requirements mandate the use of a buck or boost regulator be sure inductor used is shielded type.

2. Configuration Summary

Table 2-1. PIC32CM SG00/GC00 Family Features

Part Number	Pin Count	FLASH	SRAM	ADC	ADC Channels	Analog Comparators	TCC	SERCOM	ISO CAN_FD	USB FS	RTC	PTC	CCL	DMA Channels	WDT	Hardware Security Module Lite	PUF	FREQM	JTAG/ SWD /Trace	I/O Pins
PIC32CM5112GC00048	48	512	128	1	12	2	7	6	2	Y	Y	Y	2	8	Y	N	N	Y	Y	36
PIC32CM5112GC00064	64	512	128	1	12	2	7	6	2	Y	Y	Y	2	8	Y	N	N	Y	Y	50
PIC32CM5112GC00100	100	512	128	1	12	2	7	6	2	Y	Y	Y	2	8	Y	N	N	Y	Y	80
PIC32CM5112SG00048	48	512	128	1	12	2	7	6	2	Y	Y	Y	2	8	Y	Y	Y	Y	Y	36
PIC32CM5112SG00064	64	512	128	1	12	2	7	6	2	Y	Y	Y	2	8	Y	Y	Y	Y	Y	50
PIC32CM5112SG00100	100	512	128	1	12	2	7	6	2	Y	Y	Y	2	8	Y	Y	Y	Y	Y	80

3. Ordering Information

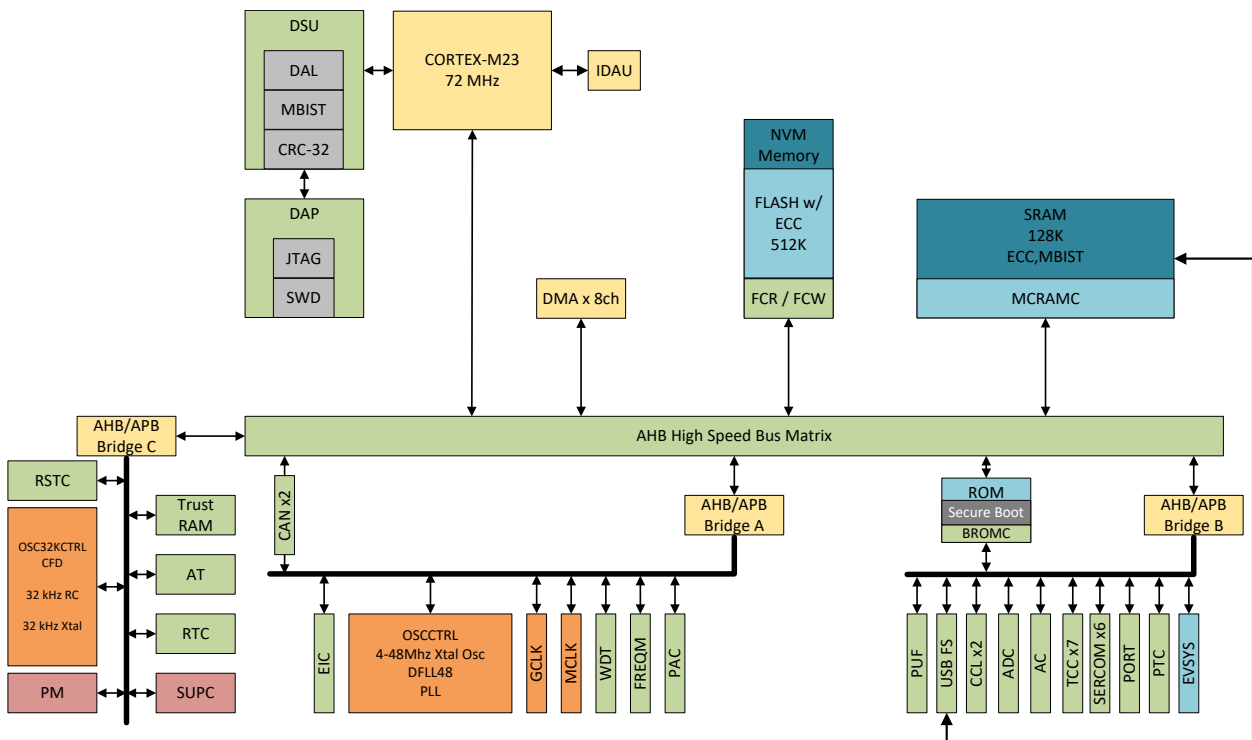


Notes:

1. Extended Temperature devices are AEC-Q100 Grade 1 Qualified.
2. The VQFN packages are with wettable flanks.

4. Block Diagram

Figure 4-1. PIC32CM SG00/GC00 Block Diagram



5. Pinout and Packaging

Each pin is controlled by the I/O Pin Controller (PORT) as a general purpose I/O and alternatively can be assigned to one of the peripheral functions: A, B, D, F, H, I, K or P.

The following tables describe the peripheral signals multiplexed to the PORT I/O pins for each package.

The column "Reset State" indicates the reset state of the line with mnemonics:

- "I/O" or "Function" indicates whether the I/O pin resets in I/O mode or in peripheral function mode.
- "I"/"O"/"Hi-Z" indicates whether the I/O is configured as an input, output, or is tri-stated.
- "PU"/"PD" indicates whether pull up, pull down, or nothing is enabled.

5.1 48-pin TQFP and 48-pin VQFN

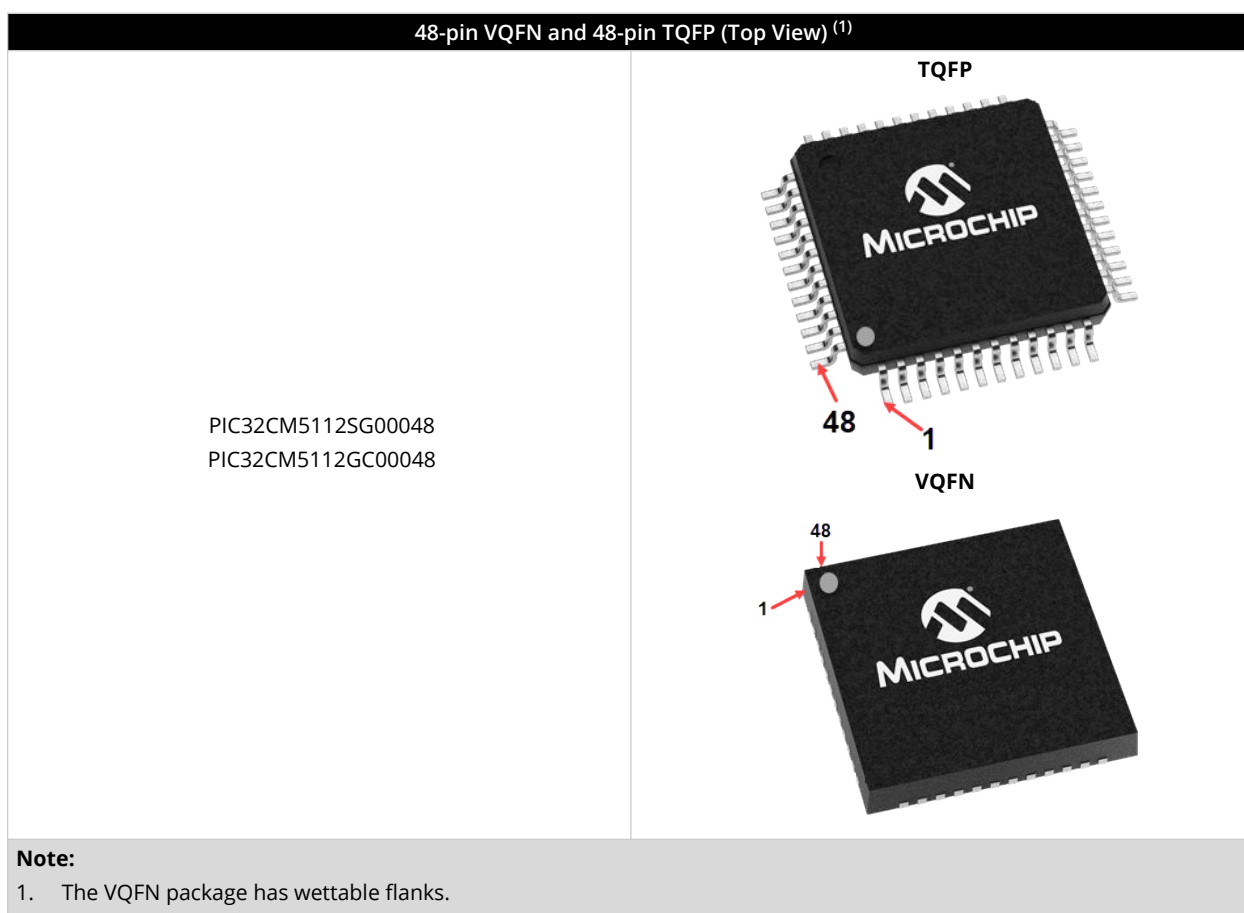


Table 5-1. 48-pin VQFN and 48-pin TQFP

Pin	MUXEN=0	Priority Function	MUXEN=1										Supply	Reset State
			PMUX Values											
			0x0	0x1			0x3	0x5	0x7	0x8	0xA	0xF		
			EIC	AC	ADC	ADCN	SERCOM	TCC	CAN	CCL	GCLK	PTC		
1	PA00	-	EXTINT[0]	-	-	-	SERCOM3/PAD[1]	TCC5/WO[1]	CAN1/RX	CCL0/IN[6]	-	XY[4]	VDDIO	Hi-Z
2	PA01	-	EXTINT[1]	CMP[0]	-	-	SERCOM3/PAD[2]	TCC6/WO[0]	CAN1/TX	CCL0/IN[7]	-	XY[5]	VDDIO	Hi-Z
3	PA02	-	EXTINT[2]	CMP[1]	-	-	SERCOM3/PAD[3]	TCC6/WO[1]	-	CCL0/OUT[2]	-	XY[6]	VDDIO	Hi-Z
4	PA03	-	EXTINT[3]	-	AIN[0]	-	SERCOM4/PAD[0]	-	-	-	-	XY[7]	VDDIO	Hi-Z
5	PA04	-	EXTINT[4]	-	AIN[1]	ANN[0]	SERCOM4/PAD[1]	-	-	-	-	XY[8]	VDDIO	Hi-Z
6	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
7	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
8	PA05	-	EXTINT[5]	-	AIN[2]	-	SERCOM4/PAD[2]	-	-	CCL1/OUT[2]	-	XY[9]	VDDIO	Hi-Z
9	PA06	-	EXTINT[6]	-	AIN[3]	ANN[2]	SERCOM4/PAD[3]	-	-	CCL1/IN[7]	-	XY[10]	VDDIO	Hi-Z
10	PA07	RTC/IN[0]	EXTINT[7]	AIN[0]	AIN[4]	-	-	-	-	-	-	XY[11]	VDDIO	Hi-Z
11	PA08	RTC/IN[1]	EXTINT[8]	AIN[1]	AIN[5]	ANN[4]	-	-	-	-	-	XY[12]	VDDIO	Hi-Z
12	PA09	RTC/IN[2]	EXTINT[9]	-	AIN[6]	VREFH	-	-	-	CCL1/IN[6]	-	XY[13]	VDDIO	Hi-Z
13	PB00	RTC/IN[3]	EXTINT[0]	AIN[2]	AIN[7]	-	-	-	-	CCL1/OUT[3]	-	XY[14]	VDDIO	Hi-Z
14	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-
15	AVSS/VSSREG	-	-	-	-	-	-	-	-	-	-	-	-	-
16	PB01	XIN32	EXTINT[1]	-	-	-	-	-	-	-	-	-	VDDIO	Hi-Z
17	PB02	XOUT32	EXTINT[2]	-	-	-	-	-	-	-	-	-	VDDIO	Hi-Z
18	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
19	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
20	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-
21	PB03	RTC/OUT[0]	EXTINT[3]	AIN[3]	AIN[8]	-	SERCOM5/PAD[0]	-	-	CCL1/IN[10]	-	XY[15]	VDDIO	Hi-Z
22	PB04	RTC/OUT[1], SUPC/OUT[0]	EXTINT[4]	-	AIN[9]	-	SERCOM5/PAD[1]	-	-	CCL1/IN[9]	-	XY[16]	VDDIO	Hi-Z
23	PB05	RTC/OUT[2], SUPC/OUT[1]	EXTINT[5]	-	AIN[10]	-	SERCOM5/PAD[2]	-	-	CCL0/IN[9]	-	XY[17]	VDDIO	Hi-Z
24	PB06	RTC/OUT[3]	EXTINT[6]	-	AIN[11]	-	SERCOM5/PAD[3]	-	-	CCL0/IN[10]	IO[0]	XY[18]	VDDIO	Hi-Z
25	PC00	-	EXTINT[0]	-	-	-	SERCOM0/PAD[0]	TCC0/WO[0]	-	CCL0/OUT[3]	IO[1]	XY[19]	VDDIO	Hi-Z
26	PC01	-	EXTINT[1]	-	-	-	SERCOM0/PAD[1]	TCC0/WO[1]	CAN0/RX	CCL0/IN[0]	IO[2]	XY[20]	VDDIO	Hi-Z
27	PC02	-	EXTINT[2]	-	-	-	SERCOM0/PAD[2]	TCC1/WO[0]	CAN0/TX	CCL0/IN[1]	IO[3]	XY[21]	VDDIO	Hi-Z
28	PC03	-	EXTINT[3]	-	-	-	SERCOM0/PAD[3]	TCC1/WO[1]	-	CCL0/OUT[0]	IO[4]	ECI[0]	VDDIO	Hi-Z
29	PC04	-	EXTINT[4]	-	-	-	SERCOM2/PAD[0]	TCC2/WO[0]	-	CCL1/IN[0]	IO[5]	ECI[1]	VDDIO	Hi-Z
30	RESET	-	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU
31	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
32	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
33	PC05	TDO	EXTINT[5]	-	-	-	SERCOM1/PAD[3]	TCC2/WO[1]	CAN1/RX	CCL1/IN[1]	IO[1]	-	VDDIO	Hi-Z
34	PC06	TDI	EXTINT[6]	-	-	-	SERCOM1/PAD[2]	-	CAN1/TX	CCL1/OUT[0]	-	-	VDDIO	Hi-Z
35	-	TMS/SWDIO	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU
36	-	TCK/SWCLK	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU
37	PD00	-	NMI	-	-	-	SERCOM1/PAD[1]	TCC3/WO[0]	-	CCL0/IN[3]	-	-	VDDIO	Hi-Z
38	PD01	-	EXTINT[1]	-	-	-	SERCOM1/PAD[0]	TCC3/WO[1]	USB/SOF_1KHZ	CCL0/IN[4]	-	-	VDDIO	Hi-Z
39	PD02	-	-	-	-	-	-	-	USB/DM	-	-	-	VDDIO	Hi-Z
40	PD03	-	-	-	-	-	-	-	USB/DP	-	-	-	VDDIO	Hi-Z
41	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
42	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
43	PD06	USERLDO	-	-	-	-	SERCOM2/PAD[1]	-	CAN0/RX	CCL1/IN[3]	-	XY[0]	VDDIO	Hi-Z
44	PD07	-	EXTINT[4]	-	-	-	SERCOM2/PAD[2]	TCC4/WO[0]	-	CCL1/IN[4]	-	XY[1]	VDDIO	Hi-Z
45	PD04	OSCCTRL/XOUT	EXTINT[2]	-	-	-	-	-	-	-	IO[1]	-	VDDIO	Hi-Z
46	PD05	OSCCTRL/XIN	EXTINT[3]	-	-	-	-	-	-	-	IO[0]	-	VDDIO	Hi-Z
47	PD08	-	EXTINT[5]	-	-	-	SERCOM2/PAD[3]	TCC4/WO[1]	CAN0/TX	CCL0/OUT[1]	-	XY[2]	VDDIO	Hi-Z
48	PD09	-	EXTINT[6]	-	-	-	SERCOM3/PAD[0]	TCC5/WO[0]	-	CCL1/OUT[1]	-	XY[3]	VDDIO	Hi-Z

5.2 64-pin TQFP and 64-pin VQFN

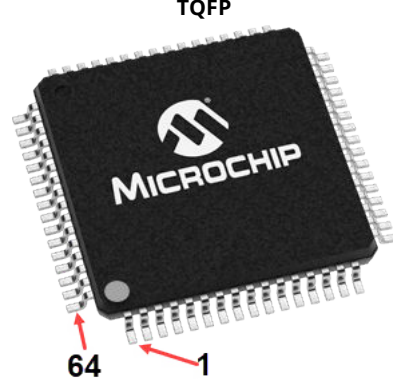
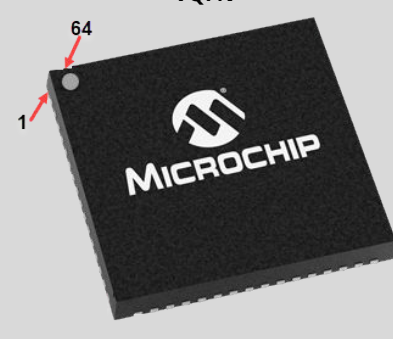
64-pin VQFN and 64-pin TQFP (Top View) ⁽¹⁾	
PIC32CM5112SG00064 PIC32CM5112GC00064	TQFP  VQFN 
Note: 1. The VQFN package has wettable flanks.	

Table 5-2. 64-pin VQFN and 64-pin TQFP

Pin	MUXEN=1	Priority Function	MUXEN=1											Supply	Reset State
			PMUX Values												
			0x0	0x1			0x3	0x5	0x7	0x8	0xA	0xF			
			EIC	AC	ADC	ADCN	SERCOM	TCC	CAN	CCL	GCLK	PTC			
1	PA00	-	EXTINT[0]	-	-	-	SERCOM3/PAD[1]	TCC5/WO[1]	CAN1/RX	CCL0/IN[6]	-	XY[4]	VDDIO	Hi-Z	
2	PA01	-	EXTINT[1]	CMP[0]	-	-	SERCOM3/PAD[2]	TCC6/WO[0]	CAN1/TX	CCL0/IN[7]	-	XY[5]	VDDIO	Hi-Z	
3	PA02	-	EXTINT[2]	CMP[1]	-	-	SERCOM3/PAD[3]	TCC6/WO[1]	-	CCL0/OUT[2]	-	XY[6]	VDDIO	Hi-Z	
4	PA10	-	EXTINT[10]	-	-	-	-	TCC4/WO[0]	CAN0/RX	CCL0/IN[8]	-	XY[22]	VDDIO	Hi-Z	
5	PA11	-	EXTINT[11]	-	-	-	-	TCC4/WO[1]	CAN0/TX	CCL1/IN[8]	-	XY[23]	VDDIO	Hi-Z	
6	PA12	-	EXTINT[12]	-	-	-	-	-	-	CCL1/IN[8]	-	XY[24]	VDDIO	Hi-Z	
7	PA03	-	EXTINT[3]	-	AIN[0]	-	SERCOM4/PAD[0]	-	-	-	-	XY[7]	VDDIO	Hi-Z	
8	PA04	-	EXTINT[4]	-	AIN[1]	ANN[0]	SERCOM4/PAD[1]	-	-	-	-	XY[8]	VDDIO	Hi-Z	
9	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	
11	PA05	-	EXTINT[5]	-	AIN[2]	-	SERCOM4/PAD[2]	-	-	CCL1/OUT[2]	-	XY[9]	VDDIO	Hi-Z	
12	PA06	-	EXTINT[6]	-	AIN[3]	ANN[2]	SERCOM4/PAD[3]	-	-	CCL1/IN[7]	-	XY[10]	VDDIO	Hi-Z	
13	PA13	-	EXTINT[13]	CMP[0]	-	-	-	-	-	-	-	XY[25]	VDDIO	Hi-Z	
14	PA07	RTC/IN[0]	EXTINT[7]	AIN[0]	AIN[4]	-	-	-	-	-	-	XY[11]	VDDIO	Hi-Z	
15	PA08	RTC/IN[1]	EXTINT[8]	AIN[1]	AIN[5]	ANN[4]	-	-	-	-	-	XY[12]	VDDIO	Hi-Z	
16	PA09	RTC/IN[2]	EXTINT[9]	-	AIN[6]	VREFH	-	-	-	CCL1/IN[6]	-	XY[13]	VDDIO	Hi-Z	
17	PB00	RTC/IN[3]	EXTINT[0]	AIN[2]	AIN[7]	-	-	-	-	CCL1/OUT[3]	-	XY[14]	VDDIO	Hi-Z	
18	PB07	RTC/IN[4]	EXTINT[8]	CMP[1]	-	-	-	-	-	CCL1/IN[11]	-	XY[26]	VDDIO	Hi-Z	
19	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-	
20	AVSS/VSSREG	-	-	-	-	-	-	-	-	-	-	-	-	-	
21	PB01	XIN32	EXTINT[1]	-	-	-	-	-	-	-	-	-	VDDIO	Hi-Z	
22	PB02	XOUT32	EXTINT[2]	-	-	-	-	-	-	-	-	-	VDDIO	Hi-Z	
23	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	
24	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	
25	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-	
26	PB03	RTC/OUT[0]	EXTINT[3]	AIN[3]	AIN[8]	-	SERCOM5/PAD[0]	-	-	CCL1/IN[10]	-	XY[15]	VDDIO	Hi-Z	
27	PB04	RTC/OUT[1], SUPC/OUT[0]	EXTINT[4]	-	AIN[9]	-	SERCOM5/PAD[1]	-	-	CCL1/IN[9]	-	XY[16]	VDDIO	Hi-Z	
28	PB05	RTC/OUT[2], SUPC/OUT[1]	EXTINT[5]	-	AIN[10]	-	SERCOM5/PAD[2]	-	-	CCL0/IN[9]	-	XY[17]	VDDIO	Hi-Z	
29	PB08	RTC/IN[5]	EXTINT[9]	-	-	-	-	TCC2/WO[1]	-	CCL1/IN[11]	-	XY[27]	VDDIO	Hi-Z	
30	PB09	RTC/OUT[4]	EXTINT[10]	-	-	-	-	TCC2/WO[0]	-	-	IO[7]	XY[28]	VDDIO	Hi-Z	
31	PB06	RTC/OUT[3]	EXTINT[6]	-	AIN[11]	-	SERCOM5/PAD[3]	-	-	CCL0/IN[10]	IO[0]	XY[18]	VDDIO	Hi-Z	
32	PB10	RTC/OUT[5]	EXTINT[11]	-	-	-	-	-	-	CCL0/IN[11]	IO[6]	XY[29]	VDDIO	Hi-Z	
33	PC00	-	EXTINT[0]	-	-	-	SERCOM0/PAD[0]	TCC0/WO[0]	-	CCL0/OUT[3]	IO[1]	XY[19]	VDDIO	Hi-Z	
34	PC01	-	EXTINT[1]	-	-	-	SERCOM0/PAD[1]	TCC0/WO[1]	CAN0/RX	CCL0/IN[0]	IO[2]	XY[20]	VDDIO	Hi-Z	
35	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	
36	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	
37	PC02	-	EXTINT[2]	-	-	-	SERCOM0/PAD[2]	TCC1/WO[0]	CAN0/TX	CCL0/IN[1]	IO[3]	XY[21]	VDDIO	Hi-Z	
38	PC03	-	EXTINT[3]	-	-	-	SERCOM0/PAD[3]	TCC1/WO[1]	-	CCL0/OUT[0]	IO[4]	ECI[0]	VDDIO	Hi-Z	
39	PC09	-	EXTINT[9]	-	-	-	-	TCC1/WO[0]	-	CCL0/IN[2]	IO[0]	XY[30]	VDDIO	Hi-Z	
40	PC04	-	EXTINT[4]	-	-	-	SERCOM2/PAD[0]	TCC2/WO[0]	-	CCL1/IN[0]	IO[5]	ECI[1]	VDDIO	Hi-Z	
41	PC10	-	EXTINT[10]	-	-	-	-	TCC1/WO[1]	-	CCL1/IN[2]	IO[1]	XY[31]	VDDIO	Hi-Z	
42	RESET	-	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU	
43	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	
44	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	
45	PC05	TDO	EXTINT[5]	-	-	-	SERCOM1/PAD[3]	TCC2/WO[1]	CAN1/RX	CCL1/IN[1]	IO[1]	-	VDDIO	Hi-Z	
46	PC06	TDI	EXTINT[6]	-	-	-	SERCOM1/PAD[2]	-	CAN1/TX	CCL1/OUT[0]	-	-	VDDIO	Hi-Z	
47	-	TMS/SWDIO	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU	
48	-	TCK/SWCLK	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU	
49	PD00	-	NMI	-	-	-	SERCOM1/PAD[1]	TCC3/WO[0]	-	CCL0/IN[3]	-	-	VDDIO	Hi-Z	
50	PD10	-	EXTINT[8]	-	-	-	SERCOM1/PAD[2]	TCC5/WO[0]	-	CCL0/IN[5]	-	-	VDDIO	Hi-Z	
51	PD11	-	EXTINT[9]	-	-	-	SERCOM1/PAD[3]	TCC5/WO[1]	-	CCL1/IN[5]	-	-	VDDIO	Hi-Z	
52	PD01	-	EXTINT[1]	-	-	-	SERCOM1/PAD[0]	TCC3/WO[1]	USB/SOF_1KHZ	CCL0/IN[4]	-	-	VDDIO	Hi-Z	
53	PD02	-	-	-	-	-	-	-	USB/DM	-	-	-	VDDIO	Hi-Z	
54	PD03	-	-	-	-	-	-	-	USB/DP	-	-	-	VDDIO	Hi-Z	
55	PD12	-	EXTINT[10]	-	-	-	-	TCC6/WO[0]	CAN1/RX	CCL1/IN[5]	-	-	VDDIO	Hi-Z	
56	PD13	-	EXTINT[11]	-	-	-	-	TCC6/WO[1]	CAN1/TX	-	-	-	VDDIO	Hi-Z	
57	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	
58	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-	
59	PD06	USERLDO	-	-	-	-	SERCOM2/PAD[1]	-	CAN0/RX	CCL1/IN[3]	-	XY[0]	VDDIO	Hi-Z	

Table 5-2. 64-pin VQFN and 64-pin TQFP (continued)

Pin	MUXEN=1	Priority Function	MUXEN=1										Supply	Reset State
			PMUX Values											
			0x0	0x1			0x3	0x5	0x7	0x8	0xA	0xF		
			EIC	AC	ADC	ADCN	SERCOM	TCC	CAN	CCL	GCLK	PTC		
60	PD07	-	EXTINT[4]	-	-	-	SERCOM2/PAD[2]	TCC4/WO[0]	-	CCL1/IN[4]	-	XY[1]	VDDIO	Hi-Z
61	PD04	OSCCTRL/XOUT	EXTINT[2]	-	-	-	-	-	-	-	IO[1]	-	VDDIO	Hi-Z
62	PD05	OSCCTRL/XIN	EXTINT[3]	-	-	-	-	-	-	-	IO[0]	-	VDDIO	Hi-Z
63	PD08	-	EXTINT[5]	-	-	-	SERCOM2/PAD[3]	TCC4/WO[1]	CAN0/TX	CCL0/OUT[1]	-	XY[2]	VDDIO	Hi-Z
64	PD09	-	EXTINT[6]	-	-	-	SERCOM3/PAD[0]	TCC5/WO[0]	-	CCL1/OUT[1]	-	XY[3]	VDDIO	Hi-Z

5.3 100-pin TQFP

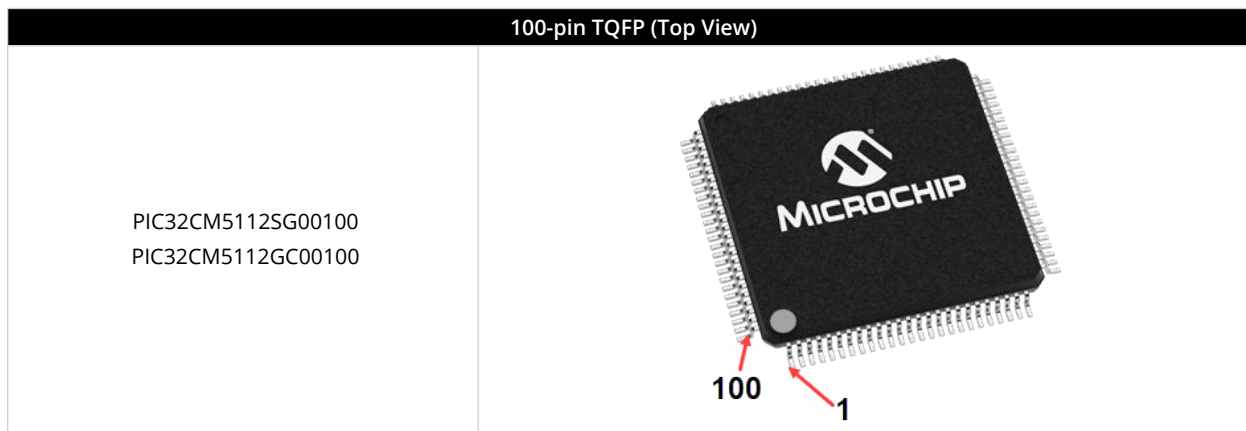


Table 5-3. 100-pin TQFP

Pin	MUXEN=0	Priority Function	MUXEN=1										Supply	Reset State
			PMUX Values											
			0x0	0x1			0x3	0x5	0x7	0x8	0xA	0xF		
			EIC	AC	ADC	ADCN	SERCOM	TCC	CAN	CCL	GCLK	PTC		
1	PA00	-	EXTINT[0]	-	-	-	SERCOM3/PAD[1]	TCC5/WO[1]	CAN1/RX	CCL0/IN[6]	-	XY[4]	VDDIO	Hi-Z
2	PA14	-	EXTINT[14]	-	-	-	SERCOM4/PAD[3]	-	-	CCL0/IN[6]	-	XY[32]	VDDIO	Hi-Z
3	PA01	-	EXTINT[1]	CMP[0]	-	-	SERCOM3/PAD[2]	TCC6/WO[0]	CAN1/TX	CCL0/IN[7]	-	XY[5]	VDDIO	Hi-Z
4	PA15	-	EXTINT[15]	-	-	-	SERCOM5/PAD[3]	-	-	CCL0/IN[7]	-	XY[33]	VDDIO	Hi-Z
5	PA02	-	EXTINT[2]	CMP[1]	-	-	SERCOM3/PAD[3]	TCC6/WO[1]	-	CCL0/OUT[2]	-	XY[6]	VDDIO	Hi-Z
6	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
7	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
8	PA10	-	EXTINT[10]	-	-	-	-	TCC4/WO[0]	CAN0/RX	CCL0/IN[8]	-	XY[22]	VDDIO	Hi-Z
9	PA11	-	EXTINT[11]	-	-	-	-	TCC4/WO[1]	CAN0/TX	CCL1/IN[8]	-	XY[23]	VDDIO	Hi-Z
10	PA12	-	EXTINT[12]	-	-	-	-	-	-	CCL1/IN[8]	-	XY[24]	VDDIO	Hi-Z
11	PA16	-	EXTINT[0]	-	-	-	SERCOM5/PAD[0]	-	-	CCL1/IN[6]	-	XY[34]	VDDIO	Hi-Z
12	PA17	-	EXTINT[1]	-	-	-	SERCOM5/PAD[1]	-	-	CCL1/IN[7]	-	XY[35]	VDDIO	Hi-Z
13	PA18	-	EXTINT[2]	-	-	-	SERCOM5/PAD[2]	-	-	CCL0/IN[8]	-	-	VDDIO	Hi-Z
14	PA03	-	EXTINT[3]	-	AIN[0]	-	SERCOM4/PAD[0]	-	-	-	-	XY[7]	VDDIO	Hi-Z
15	PA04	-	EXTINT[4]	-	AIN[1]	ANN[0]	SERCOM4/PAD[1]	-	-	-	-	XY[8]	VDDIO	Hi-Z
16	PA05	-	EXTINT[5]	-	AIN[2]	-	SERCOM4/PAD[2]	-	-	CCL1/OUT[2]	-	XY[9]	VDDIO	Hi-Z
17	PA06	-	EXTINT[6]	-	AIN[3]	ANN[2]	SERCOM4/PAD[3]	-	-	CCL1/IN[7]	-	XY[10]	VDDIO	Hi-Z
18	PA13	-	EXTINT[13]	CMP[0]	-	-	-	-	-	-	-	XY[25]	VDDIO	Hi-Z
19	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
20	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
21	PA07	RTC/IN[0]	EXTINT[7]	AIN[0]	AIN[4]	-	-	-	-	-	-	XY[11]	VDDIO	Hi-Z
22	PA08	RTC/IN[1]	EXTINT[8]	AIN[1]	AIN[5]	ANN[4]	-	-	-	-	-	XY[12]	VDDIO	Hi-Z
23	PA09	RTC/IN[2]	EXTINT[9]	-	AIN[6]	VREFH	-	-	-	CCL1/IN[6]	-	XY[13]	VDDIO	Hi-Z
24	PA20	-	EXTINT[4]	-	-	-	-	TCC0/WO[1]	-	CCL1/OUT[2]	-	-	VDDIO	Hi-Z
25	PA19	-	EXTINT[3]	-	-	-	-	TCC0/WO[0]	-	CCL0/OUT[2]	-	-	VDDIO	Hi-Z
26	PB00	RTC/IN[3]	EXTINT[0]	AIN[2]	AIN[7]	-	-	-	-	CCL1/OUT[3]	-	XY[14]	VDDIO	Hi-Z
27	PB07	RTC/IN[4]	EXTINT[8]	CMP[1]	-	-	-	-	-	CCL1/IN[11]	-	XY[26]	VDDIO	Hi-Z
28	AVDD	-	-	-	-	-	-	-	-	-	-	-	-	-
29	AVSS/VSSREG	-	-	-	-	-	-	-	-	-	-	-	-	-
30	PB01	XIN32	EXTINT[1]	-	-	-	-	-	-	-	-	-	VDDIO	Hi-Z
31	PB02	XOUT32	EXTINT[2]	-	-	-	-	-	-	-	-	-	VDDIO	Hi-Z
32	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
33	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
34	VDDREG	-	-	-	-	-	-	-	-	-	-	-	-	-
35	PB11	RTC/IN[6]	EXTINT[12]	-	-	-	-	TCC0/WO[0]	-	CCL1/IN[9]	-	-	VDDIO	Hi-Z
36	PB12	RTC/IN[7]	EXTINT[2]	-	-	-	-	TCC0/WO[1]	-	CCL0/IN[9]	-	-	VDDIO	Hi-Z
37	PB13	RTC/OUT[6]	EXTINT[14]	-	-	-	-	TCC3/WO[0]	-	CCL1/IN[10]	-	-	VDDIO	Hi-Z
38	PB14	RTC/OUT[7]	EXTINT[15]	-	-	-	-	TCC3/WO[1]	-	CCL0/IN[10]	-	-	VDDIO	Hi-Z
39	PB03	RTC/OUT[0]	EXTINT[3]	AIN[3]	AIN[8]	-	SERCOM5/PAD[0]	-	-	CCL1/IN[10]	-	XY[15]	VDDIO	Hi-Z
40	PB04	RTC/OUT[1], SUPC/OUT[0]	EXTINT[4]	-	AIN[9]	-	SERCOM5/PAD[1]	-	-	CCL1/IN[9]	-	XY[16]	VDDIO	Hi-Z

Table 5-3. 100-pin TQFP (continued)

Pin	MUXEN=0	Priority Function	MUXEN=1										Supply	Reset State
			PMUX Values											
			0x0	0x1			0x3	0x5	0x7	0x8	0xA	0xF		
			EIC	AC	ADC	ADCN	SERCOM	TCC	CAN	CCL	GCLK	PTC		
41	PB05	RTC/OUT[2], SUPC/OUT[1]	EXTINT[5]	-	AIN[10]	-	SERCOM5/PAD[2]	-	-	CCL0/IN[9]	-	XY[17]	VDDIO	Hi-Z
42	PB08	RTC/IN[5]	EXTINT[9]	-	-	-	-	TCC2/WO[1]	-	CCL1/IN[11]	-	XY[27]	VDDIO	Hi-Z
43	PB09	RTC/OUT[4]	EXTINT[10]	-	-	-	-	TCC2/WO[0]	-	-	IO[7]	XY[28]	VDDIO	Hi-Z
44	PB06	RTC/OUT[3]	EXTINT[6]	-	AIN[11]	-	SERCOM5/PAD[3]	-	-	CCL0/IN[10]	IO[0]	XY[18]	VDDIO	Hi-Z
45	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
46	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
47	PB15	-	EXTINT[0]	-	-	-	SERCOM0/PAD[0]	-	-	CCL0/IN[11]	IO[0]	-	VDDIO	Hi-Z
48	PB16	-	EXTINT[1]	-	-	-	SERCOM0/PAD[1]	-	-	CCL0/OUT[3]	IO[6]	-	VDDIO	Hi-Z
49	PB17	-	EXTINT[2]	-	-	-	SERCOM0/PAD[2]	-	-	CCL1/OUT[3]	-	-	VDDIO	Hi-Z
50	PB10	RTC/OUT[5]	EXTINT[11]	-	-	-	-	-	-	CCL0/IN[11]	IO[6]	XY[29]	VDDIO	Hi-Z
51	PC00	-	EXTINT[0]	-	-	-	SERCOM0/PAD[0]	TCC0/WO[0]	-	CCL0/OUT[3]	IO[1]	XY[19]	VDDIO	Hi-Z
52	PC11	-	EXTINT[11]	-	-	-	SERCOM0/PAD[3]	-	-	CCL1/IN[0]	-	-	VDDIO	Hi-Z
53	PC12	-	EXTINT[12]	-	-	-	SERCOM1/PAD[3]	-	-	CCL0/IN[0]	-	-	VDDIO	Hi-Z
54	PC01	-	EXTINT[1]	-	-	-	SERCOM0/PAD[1]	TCC0/WO[1]	CAN0/RX	CCL0/IN[0]	IO[2]	XY[20]	VDDIO	Hi-Z
55	PC13	-	EXTINT[13]	-	-	-	SERCOM1/PAD[2]	-	-	CCL0/OUT[0]	-	-	VDDIO	Hi-Z
56	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
57	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
58	PC02	-	EXTINT[2]	-	-	-	SERCOM0/PAD[2]	TCC1/WO[0]	CAN0/TX	CCL0/IN[1]	IO[3]	XY[21]	VDDIO	Hi-Z
59	PC14	-	EXTINT[14]	-	-	-	SERCOM1/PAD[1]	-	-	CCL0/IN[1]	-	-	VDDIO	Hi-Z
60	PC15	-	EXTINT[15]	-	-	-	SERCOM1/PAD[0]	-	-	CCL0/IN[2]	-	-	VDDIO	Hi-Z
61	PC03	-	EXTINT[3]	-	-	-	SERCOM0/PAD[3]	TCC1/WO[1]	-	CCL0/OUT[0]	IO[4]	ECI[0]	VDDIO	Hi-Z
62	PC16	-	EXTINT[0]	-	-	-	SERCOM2/PAD[3]	-	-	CCL1/IN[2]	-	-	VDDIO	Hi-Z
63	PC09	-	EXTINT[9]	-	-	-	-	TCC1/WO[0]	-	CCL0/IN[2]	IO[0]	XY[30]	VDDIO	Hi-Z
64	PC04	-	EXTINT[4]	-	-	-	SERCOM2/PAD[0]	TCC2/WO[0]	-	CCL1/IN[0]	IO[5]	ECI[1]	VDDIO	Hi-Z
65	PC17	-	EXTINT[1]	-	-	-	SERCOM2/PAD[2]	-	-	CCL1/IN[1]	-	-	VDDIO	Hi-Z
66	PC10	-	EXTINT[10]	-	-	-	-	TCC1/WO[1]	-	CCL1/IN[2]	IO[1]	XY[31]	VDDIO	Hi-Z
67	RESET	-	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU
68	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
69	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
70	PC18	-	EXTINT[2]	-	-	-	SERCOM2/PAD[1]	-	-	-	-	-	VDDIO	Hi-Z
71	PC19	-	EXTINT[3]	-	-	-	SERCOM2/PAD[0]	-	-	CCL1/OUT[0]	-	-	VDDIO	Hi-Z
72	PC05	TDO	EXTINT[5]	-	-	-	SERCOM1/PAD[3]	TCC2/WO[1]	CAN1/RX	CCL1/IN[1]	IO[1]	-	VDDIO	Hi-Z
73	PC06	TDI	EXTINT[6]	-	-	-	SERCOM1/PAD[2]	-	CAN1/TX	CCL1/OUT[0]	-	-	VDDIO	Hi-Z
74	-	TMS/SWDIO	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU
75	-	TCK/SWCLK	-	-	-	-	-	-	-	-	-	-	VDDIO	I - PU
76	PD00	-	NMI	-	-	-	SERCOM1/PAD[1]	TCC3/WO[0]	-	CCL0/IN[3]	-	-	VDDIO	Hi-Z
77	PD10	-	EXTINT[8]	-	-	-	SERCOM1/PAD[2]	TCC5/WO[0]	-	CCL0/IN[5]	-	-	VDDIO	Hi-Z
78	PD14	-	EXTINT[12]	-	-	-	SERCOM3/PAD[3]	-	-	CCL0/OUT[1]	-	-	VDDIO	Hi-Z
79	PD11	-	EXTINT[9]	-	-	-	SERCOM1/PAD[3]	TCC5/WO[1]	-	CCL1/IN[5]	-	-	VDDIO	Hi-Z
80	PD01	-	EXTINT[1]	-	-	-	SERCOM1/PAD[0]	TCC3/WO[1]	USB/SOF_1KHZ	CCL0/IN[4]	-	-	VDDIO	Hi-Z
81	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
82	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
83	PD02	-	-	-	-	-	-	-	USB/DM	-	-	-	VDDIO	Hi-Z
84	PD03	-	-	-	-	-	-	-	USB/DP	-	-	-	VDDIO	Hi-Z
85	PD15	-	EXTINT[13]	-	-	-	SERCOM3/PAD[1]	-	-	CCL1/OUT[1]	-	-	VDDIO	Hi-Z
86	PD16	-	EXTINT[14]	-	-	-	SERCOM3/PAD[0]	-	-	CCL1/IN[4]	-	-	VDDIO	Hi-Z
87	PD17	-	EXTINT[15]	-	-	-	SERCOM3/PAD[2]	-	-	CCL0/IN[4]	-	-	VDDIO	Hi-Z
88	PD12	-	EXTINT[10]	-	-	-	-	TCC6/WO[0]	CAN1/RX	CCL1/IN[5]	-	-	VDDIO	Hi-Z
89	PD18	-	EXTINT[0]	-	-	-	SERCOM4/PAD[2]	-	-	CCL0/IN[3]	-	-	VDDIO	Hi-Z
90	PD13	-	EXTINT[11]	-	-	-	-	TCC6/WO[1]	CAN1/TX	-	-	-	VDDIO	Hi-Z
91	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-
92	VDDIO	-	-	-	-	-	-	-	-	-	-	-	-	-
93	PD06	USERLDO	-	-	-	-	SERCOM2/PAD[1]	-	CAN0/RX	CCL1/IN[3]	-	XY[0]	VDDIO	Hi-Z
94	PD07	-	EXTINT[4]	-	-	-	SERCOM2/PAD[2]	TCC4/WO[0]	-	CCL1/IN[4]	-	XY[1]	VDDIO	Hi-Z
95	PD19	-	EXTINT[1]	-	-	-	SERCOM4/PAD[0]	-	-	CCL0/IN[5]	-	-	VDDIO	Hi-Z
96	PD04	OSCCTRL/XOUT	EXTINT[2]	-	-	-	-	-	-	-	IO[1]	-	VDDIO	Hi-Z
97	PD05	OSCCTRL/XIN	EXTINT[3]	-	-	-	-	-	-	-	IO[0]	-	VDDIO	Hi-Z
98	PD08	-	EXTINT[5]	-	-	-	SERCOM2/PAD[3]	TCC4/WO[1]	CAN0/TX	CCL0/OUT[1]	-	XY[2]	VDDIO	Hi-Z
99	PD20	-	EXTINT[2]	-	-	-	SERCOM4/PAD[1]	-	-	CCL1/IN[3]	-	-	VDDIO	Hi-Z

Table 5-3. 100-pin TQFP (continued)

Pin	MUXEN=0	Priority Function	MUXEN=1										Supply	Reset State
			PMUX Values											
			0x0	0x1			0x3	0x5	0x7	0x8	0xA	0xF		
			EIC	AC	ADC	ADCN	SERCOM	TCC	CAN	CCL	GCLK	PTC		
100	PD09	-	EXTINT[6]	-	-	-	SERCOM3/PAD[0]	TCC5/WO[0]	-	CCL1/OUT[1]	-	XY[3]	VDDIO	Hi-Z

6. Signal Description

Table 6-1. AC Signal Descriptions

PIN NAME	PIN NUMBER			I/O Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
AC_AIN0	21	14	10	—	I	Analog	Comparator 0 & 1 external analog inputs
AC_AIN1	22	15	11	—	I	Analog	
AC_AIN2	26	17	13	—	I	Analog	
AC_AIN3	39	26	21	—	I	Analog	
AC_CMP0	3	2	2	1	O	Analog	Comparator 0 Output
AC_CMP0	18	13	—	2	O	Analog	
AC_CMP1	5	3	3	1	O	Analog	Comparator 1 Output
AC_CMP1	27	18	—	2	O	Analog	

Table 6-2. ADC Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
ADC_VINP0	14	7	4	I	Analog	ADC Module 0 analog inputs
ADC_VINP1	15	8	5	I	Analog	
ADC_VINP2	16	11	8	I	Analog	
ADC_VINP3	17	12	9	I	Analog	
ADC_VINP4	21	14	10	I	Analog	
ADC_VINP5	22	15	11	I	Analog	
ADC_VINP6	23	16	12	I	Analog	
ADC_VINP7	26	17	13	I	Analog	
ADC_VINP8	39	26	21	I	Analog	
ADC_VINP9	40	27	22	I	Analog	
ADC_VINP10	41	28	23	I	Analog	
ADC_VINP11	44	31	17	I	Analog	
ADC_VINM0	15	8	5	I	Analog	
ADC_VINM2	17	12	9	I	Analog	
ADC_VINM4	22	15	11	I	Analog	
ADC_VREFH	23	16	12	I	Analog	

Table 6-3. Programming-Debugging Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
TCK_SWCLK	75	48	36	I	ST	Debug Serial Wire Input Clock
TDI	73	46	34	I	ST	Debug/Scan JTAG Data Input
TDO	72	45	33	O	CMOS	Debug/Scan JTAG Data Out
TMS_SWDIO	74	47	35	O	CMOS	Debug Serial Wire Data In/Out
RESET	67	42	30	I	ST	MCU Reset input (Active Low)

Table 6-4. CAN Signal Descriptions

PIN NAME	PIN NUMBER			I/O Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
CAN0_RX	8	4	—	3	I	ST	CAN0_FD Receive
CAN0_RX	54	34	26	1			
CAN0_RX	93	—	—	2			
CAN0_TX	9	5	—	3	O	CMOS	CAN0_FD Transmit
CAN0_TX	58	37	27	1			
CAN0_TX	98	63	—	2			
CAN1_RX	1	1	1	2	I	ST	CAN1_FD Receive
CAN1_RX	72	45	33	1			
CAN1_RX	88	55	—	3			
CAN1_TX	3	2	2	2	O	CMOS	CAN1_FD Transmit
CAN1_TX	73	46	34	1			
CAN1_TX	90	56	—	3			

Table 6-5. CCL Signal Descriptions

PIN NAME	PIN NUMBER			IO Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
CCL0_IN0	53	—	—	2	I	Digital Input	CCL0 Lookup Table Input 0
CCL0_IN0	54	34	26	1			
CCL0_IN1	58	37	27	1	I	Digital Input	CCL0 Lookup Table Input 1
CCL0_IN1	59	—	—	2			
CCL0_IN10	38	—	—	2	I	Digital Input	CCL0 Lookup Table Input 2
CCL0_IN10	44	31	24	1			
CCL0_IN11	47	—	—	1	I	Digital Input	CCL0 Lookup Table Input 3
CCL0_IN11	50	32	—	2			
CCL0_IN2	60	—	—	1	I	Digital Input	CCL0 Lookup Table Input 4
CCL0_IN2	63	39	—	2			
CCL0_IN3	76	49	37	1	I	Digital Input	CCL0 Lookup Table Input 5
CCL0_IN3	89	—	—	2			
CCL0_IN4	80	52	38	1	I	Digital Input	CCL0 Lookup Table Input 6
CCL0_IN4	87	—	—	2			
CCL0_IN5	77	50	—	1	I	Digital Input	CCL0 Lookup Table Input 7
CCL0_IN5	95	61	45	2			
CCL0_IN6	1	1	1	1	I	Digital Input	CCL0 Lookup Table Input 8
CCL0_IN6	2	—	—	2			
CCL0_IN7	3	2	2	2	I	Digital Input	CCL0 Lookup Table Input 9
CCL0_IN7	4	—	—	1			
CCL0_IN8	8	4	—	2	I	Digital Input	CCL0 Lookup Table Input 10
CCL0_IN8	13	—	—	1			
CCL0_IN9	36	—	—	2	I	Digital Input	CCL0 Lookup Table Input 11
CCL0_IN9	41	28	23	1			
CCL0_OUT0	55	—	—	2	O	Digital Output	CCL0 Lookup Table output 0
CCL0_OUT0	61	38	28	1			
CCL0_OUT1	78	—	—	2	O	Digital Output	CCL0 Lookup Table output 1
CCL0_OUT1	98	63	47	1			

Table 6-5. CCL Signal Descriptions (continued)

PIN NAME	PIN NUMBER			IO Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
CCL0_OUT2	5	3	3	1	O	Digital Output	CCL0 Lookup Table output 2
CCL0_OUT2	25	—	—	2			
CCL0_OUT3	48	—	—	2	O	Digital Output	CCL0 Lookup Table output 3
CCL0_OUT3	51	33	25	1			
CCL1_IN0	52	—	—	2	I	Digital Input	CCL1 Lookup Table Input 0
CCL1_IN0	64	40	29	1			
CCL1_IN1	65	—	—	2	I	Digital Input	CCL1 Lookup Table Input 1
CCL1_IN1	72	45	33	1			
CCL1_IN10	37	—	—	2	I	Digital Input	CCL1 Lookup Table Input 2
CCL1_IN10	39	26	21	1			
CCL1_IN11	27	18	—	1	I	Digital Input	CCL1 Lookup Table Input 3
CCL1_IN11	42	29	—	2			
CCL1_IN2	62	—	—	2	I	Digital Input	CCL1 Lookup Table Input 4
CCL1_IN2	66	41	—	1			
CCL1_IN3	93	59	43	1	I	Digital Input	CCL1 Lookup Table Input 5
CCL1_IN3	99	—	—	2			
CCL1_IN4	86	—	—	2	I	Digital Input	CCL1 Lookup Table Input 6
CCL1_IN4	94	60	44	1			
CCL1_IN5	79	51	—	2	I	Digital Input	CCL1 Lookup Table Input 7
CCL1_IN5	88	55	—	1			
CCL1_IN6	11	—	—	1	I	Digital Input	CCL1 Lookup Table Input 8
CCL1_IN6	23	16	12	2			
CCL1_IN7	12	—	—	2	I	Digital Input	CCL1 Lookup Table Input 9
CCL1_IN7	17	12	9	1			
CCL1_IN8	9	5	—	2	I	Digital Input	CCL1 Lookup Table Input 10
CCL1_IN8	10	6	—	1			
CCL1_IN9	35	—	—	2	I	Digital Input	CCL1 Lookup Table Input 11
CCL1_IN9	40	27	22	1			
CCL1_OUT0	71	—	—	2	O	Digital Output	CCL1 Lookup Table output 0
CCL1_OUT0	73	46	34	1			
CCL1_OUT1	85	—	—	2	O	Digital Output	CCL1 Lookup Table output 1
CCL1_OUT1	100	64	48	1			
CCL1_OUT2	16	11	8	1	O	Digital Output	CCL1 Lookup Table output 2
CCL1_OUT2	24	—	—	2			
CCL1_OUT3	26	17	13	1	O	Digital Output	CCL1 Lookup Table output 3
CCL1_OUT3	49	—	—	2			

Table 6-6. EIC Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
EIC_EXTINT0	1/11/26/47/51/62/89	1/17/1933	1/13/2025	I	ST	External Interrupt 0 input
EIC_EXTINT1	3/12/30/48/54/65/80/95	2/21/34/52	2/16/26/38	I	ST	External Interrupt 1 input
EIC_EXTINT2	5/13/31/36/49/58/70/96/99	3/22/37/61	3/17/27/45	I	ST	External Interrupt 2 input
EIC_EXTINT3	14/25/39/61/71/97	7/26/38/62	4/21/28/46	I	ST	External Interrupt 3 input

Table 6-6. EIC Signal Descriptions (continued)

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
EIC_EXTINT4	15/24/40/64/94	8/27/40/60	5/22/29/44	I	ST	External Interrupt 4 input
EIC_EXTINT5	16/41/72/98	11/28/45/63	8/23/33/47	I	ST	External Interrupt 5 input
EIC_EXTINT6	17/44/73/100	12/31/46/64	9/24/34/48	I	ST	External Interrupt 6 input
EIC_EXTINT7	21	14	10	I	ST	External Interrupt 7 input
EIC_EXTINT8	22/27/77	15/18/50	11	I	ST	External Interrupt 8 input
EIC_EXTINT9	23/42/63/79	46/29/39/51	12	I	ST	External Interrupt 9 input
EIC_EXTINT10	8/43/66/88	4/30/41/55	—	I	ST	External Interrupt 10 input
EIC_EXTINT11	9/50/52/90	5/32/56	—	I	ST	External Interrupt 11 input
EIC_EXTINT12	10/35/53/78	6	—	I	ST	External Interrupt 12 input
EIC_EXTINT13	18/55/85	—	—	I	ST	External Interrupt 13 input
EIC_EXTINT14	2/37/59/86	—	—	I	ST	External Interrupt 14 input
EIC_EXTINT15	4/38/60/87	—	—	I	ST	External Interrupt 15 input
EIC_NMI	76	49	37	I	ST	Non-Maskable Interrupt input

Table 6-7. GCLK Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
GCLK_IO0	44/47/63/97	31/39/62	24/46	I	ST	Ext GCLK[7:0] I/O pin Input clock source
GCLK_IO1	51/66/72/96	33/41/45/61	25/33/45	I	ST	
GCLK_IO2	54	34	26	I	ST	
GCLK_IO3	58	37	27	I	ST	
GCLK_IO4	61	38	28	I	ST	
GCLK_IO5	64	40	29	I	ST	
GCLK_IO6	48/50	32	—	I	ST	
GCLK_IO7	43	30	—	I	ST	
XIN32	30	21	16	I	Analog	32.768Khz oscillator input
XOUT32	31	22	17	O	Analog	32.768Khz oscillator output
XIN	97	62	46	I	Analog	Primary oscillator input
XOUT	96	61	45	O	Analog	Primary oscillator output

Table 6-8. PORT Signals Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
PA00	1	1	1	I/O	ST/CMOS	I/O Port A Pins
PA01	3	2	2	I/O	ST/CMOS	
PA02	5	3	3	I/O	ST/CMOS	
PA03	14	7	4	I/O	ST/CMOS	
PA04	15	8	5	I/O	ST/CMOS	
PA05	16	11	8	I/O	ST/CMOS	
PA06	17	12	9	I/O	ST/CMOS	
PA07	21	14	10	I/O	ST/CMOS	
PA08	22	15	11	I/O	ST/CMOS	
PA09	23	16	12	I/O	ST/CMOS	
PA10	8	4	—	I/O	ST/CMOS	
PA11	9	5	—	I/O	ST/CMOS	
PA12	10	6	—	I/O	ST/CMOS	
PA13	18	13	—	I/O	ST/CMOS	
PA14	2	—	—	I/O	ST/CMOS	
PA15	4	—	—	I/O	ST/CMOS	
PA16	11	—	—	I/O	ST/CMOS	
PA17	12	—	—	I/O	ST/CMOS	
PA18	13	—	—	I/O	ST/CMOS	
PA19	25	—	—	I/O	ST/CMOS	
PA20	24	—	—	I/O	ST/CMOS	
PB00	26	17	13	I/O	ST/CMOS	I/O Port B Pins
PB01	30	21	16	I/O	ST/CMOS	
PB02	31	22	17	I/O	ST/CMOS	
PB03	39	26	21	I/O	ST/CMOS	
PB04	40	27	22	I/O	ST/CMOS	
PB05	41	28	23	I/O	ST/CMOS	
PB06	44	31	24	I/O	ST/CMOS	
PB07	27	18	—	I/O	ST/CMOS	
PB08	42	29	—	I/O	ST/CMOS	
PB09	43	30	—	I/O	ST/CMOS	
PB10	50	32	—	I/O	ST/CMOS	
PB11	35	—	—	I/O	ST/CMOS	
PB12	36	—	—	I/O	ST/CMOS	
PB13	37	—	—	I/O	ST/CMOS	
PB14	38	—	—	I/O	ST/CMOS	
PB15	47	—	—	I/O	ST/CMOS	
PB16	48	—	—	I/O	ST/CMOS	
PB17	49	—	—	I/O	ST/CMOS	

Table 6-8. PORT Signals Descriptions (continued)

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
PC00	51	33	25	I/O	ST/CMOS	I/O Port C Pins
PC01	54	34	26	I/O	ST/CMOS	
PC02	58	37	27	I/O	ST/CMOS	
PC03	61	38	28	I/O	ST/CMOS	
PC04	64	40	29	I/O	ST/CMOS	
PC05	72	45	33	I/O	ST/CMOS	
PC06	73	46	34	I/O	ST/CMOS	
PC07	74	47	35	I/O	ST/CMOS	
PC08	75	48	36	I/O	ST/CMOS	
PC09	63	39	—	I/O	ST/CMOS	
PC10	66	41	—	I/O	ST/CMOS	
PC11	52	—	—	I/O	ST/CMOS	
PC12	53	—	—	I/O	ST/CMOS	
PC13	55	—	—	I/O	ST/CMOS	
PC14	59	—	—	I/O	ST/CMOS	
PC15	60	—	—	I/O	ST/CMOS	
PC16	62	—	—	I/O	ST/CMOS	
PC17	65	—	—	I/O	ST/CMOS	
PC18	70	—	—	I/O	ST/CMOS	
PC19	71	—	—	I/O	ST/CMOS	
PD00	76	49	37	I/O	ST/CMOS	I/O Port D Pins
PD01	80	52	38	I/O	ST/CMOS	
PD02	83	53	39	I/O	ST/CMOS	
PD03	84	54	40	I/O	ST/CMOS	
PD04	96	61	45	I/O	ST/CMOS	
PD05	97	62	46	I/O	ST/CMOS	
PD06	93	59	43	I/O	ST/CMOS	
PD07	94	60	44	I/O	ST/CMOS	
PD08	98	63	47	I/O	ST/CMOS	
PD09	100	64	48	I/O	ST/CMOS	
PD10	77	50	—	I/O	ST/CMOS	
PD11	79	51	—	I/O	ST/CMOS	
PD12	88	55	—	I/O	ST/CMOS	
PD13	90	56	—	I/O	ST/CMOS	
PD14	78	—	—	I/O	ST/CMOS	
PD15	85	—	—	I/O	ST/CMOS	
PD16	86	—	—	I/O	ST/CMOS	
PD17	87	—	—	I/O	ST/CMOS	
PD18	89	—	—	I/O	ST/CMOS	
PD19	95	—	—	I/O	ST/CMOS	
PD20	99	—	—	I/O	ST/CMOS	

Table 6-9. PTC Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
XY0	93	59	43	I/O	ST/CMOS	Digital Output (X-line) /Analog I/O (Y-line)
XY1	94	60	44	I/O	ST/CMOS	
XY2	98	63	47	I/O	ST/CMOS	
XY3	100	64	48	I/O	ST/CMOS	
XY4	1	1	1	I/O	ST/CMOS	
XY5	3	2	2	I/O	ST/CMOS	
XY6	5	3	3	I/O	ST/CMOS	
XY7	14	7	4	I/O	ST/CMOS	
XY8	15	8	5	I/O	ST/CMOS	
XY9	16	11	8	I/O	ST/CMOS	
XY10	17	12	9	I/O	ST/CMOS	
XY11	21	14	10	I/O	ST/CMOS	
XY12	22	15	11	I/O	ST/CMOS	
XY13	23	16	12	I/O	ST/CMOS	
XY14	26	17	13	I/O	ST/CMOS	
XY15	39	26	21	I/O	ST/CMOS	
XY16	40	27	22	I/O	ST/CMOS	
XY17	41	28	23	I/O	ST/CMOS	
XY18	44	31	24	I/O	ST/CMOS	
XY19	51	33	25	I/O	ST/CMOS	
XY20	54	34	26	I/O	ST/CMOS	
XY21	58	37	27	I/O	ST/CMOS	
XY22	8	4	—	I/O	ST/CMOS	
XY23	9	5	—	I/O	ST/CMOS	
XY24	10	6	—	I/O	ST/CMOS	
XY25	18	13	—	I/O	ST/CMOS	
XY26	27	18	—	I/O	ST/CMOS	
XY27	42	29	—	I/O	ST/CMOS	
XY28	43	30	—	I/O	ST/CMOS	
XY29	50	32	—	I/O	ST/CMOS	
XY30	63	39	—	I/O	ST/CMOS	
XY31	66	41	—	I/O	ST/CMOS	
XY32	2	—	—	I/O	ST/CMOS	
XY33	4	—	—	I/O	ST/CMOS	
XY34	11	—	—	I/O	ST/CMOS	
XY35	12	—	—	I/O	ST/CMOS	
ECI0	61	38	28	I	Analog	External Capacitor Analog Input 0
ECI1	64	41	29	I	Analog	External Capacitor Analog Input 1

Table 6-10. RTC Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
IN0	21	14	10	I	ST	RTC security tamper input loop(s)
IN1	22	15	11	I	ST	
IN2	23	16	12	I	ST	
IN3	26	17	13	I	ST	
IN4	27	18	—	I	ST	
IN5	42	29	—	I	ST	
IN6	35	—	—	I	ST	
IN7	36	—	—	I	ST	RTC security tamper output loop(s)
OUT0	39	26	21	O	CMOS	
OUT1	40	27	22	O	CMOS	
OUT2	41	28	23	O	CMOS	
OUT3	44	31	24	O	CMOS	
OUT4	43	30	—	O	CMOS	
OUT5	50	32	—	O	CMOS	
OUT6	37	—	—	O	CMOS	
OUT7	38	—	—	O	CMOS	

Table 6-11. SERCOM Signal Descriptions

PIN NAME	PIN NUMBER			I/O Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
SERCOM0_PAD0	47	—	—	2	I/O	ST/CMOS	SERCOM 0 I/O Pads
SERCOM0_PAD0	51	33	25	1	I/O	ST/CMOS	
SERCOM0_PAD1	48	—	—	2	I/O	ST/CMOS	
SERCOM0_PAD1	54	34	26	1	I/O	ST/CMOS	
SERCOM0_PAD2	49	—	—	2	I/O	ST/CMOS	
SERCOM0_PAD2	58	37	27	1	I/O	ST/CMOS	
SERCOM0_PAD3	52	—	—	2	I/O	ST/CMOS	
SERCOM0_PAD3	61	38	28	1	I/O	ST/CMOS	
SERCOM1_PAD0	60/80	52	38	1	I/O	ST/CMOS	SERCOM 1 I/O Pads
SERCOM1_PAD1	59/76	49	37	1	I/O	ST/CMOS	
SERCOM1_PAD2	55	46	—	2	I/O	ST/CMOS	
SERCOM1_PAD2	73	50	34	1	I/O	ST/CMOS	
SERCOM1_PAD2	77	45	—	1	I/O	ST/CMOS	
SERCOM1_PAD3	53	51	—	2	I/O	ST/CMOS	
SERCOM1_PAD3	72	45	33	1	I/O	ST/CMOS	
SERCOM1_PAD3	79	51	—	2	I/O	ST/CMOS	SERCOM 2 I/O Pads
SERCOM2_PAD0	64	40	29	1	I/O	ST/CMOS	
SERCOM2_PAD0	71	—	—	2	I/O	ST/CMOS	
SERCOM2_PAD1	70	—	—	2	I/O	ST/CMOS	
SERCOM2_PAD1	93	59	59	1	I/O	ST/CMOS	
SERCOM2_PAD2	65	—	—	2	I/O	ST/CMOS	
SERCOM2_PAD2	94	60	44	1	I/O	ST/CMOS	
SERCOM2_PAD3	62	—	—	2	I/O	ST/CMOS	
SERCOM2_PAD3	98	63	47	1	I/O	ST/CMOS	

Table 6-11. SERCOM Signal Descriptions (continued)

PIN NAME	PIN NUMBER			I/O Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
SERCOM3_PAD0	86	—	—	2	I/O	ST/CMOS	SERCOM 3 I/O Pads
SERCOM3_PAD0	100	64	48	1	I/O	ST/CMOS	
SERCOM3_PAD1	1	—	—	1	I/O	ST/CMOS	
SERCOM3_PAD1	85	1	1	2	I/O	ST/CMOS	
SERCOM3_PAD2	3	—	—	1	I/O	ST/CMOS	
SERCOM3_PAD2	87	2	2	2	I/O	ST/CMOS	
SERCOM3_PAD3	5	—	—	1	I/O	ST/CMOS	
SERCOM3_PAD3	78	3	3	2	I/O	ST/CMOS	
SERCOM4_PAD0	14	—	—	1	I/O	ST/CMOS	SERCOM 4 I/O Pads
SERCOM4_PAD0	95	7	4	2	I/O	ST/CMOS	
SERCOM4_PAD1	15	—	—	1	I/O	ST/CMOS	
SERCOM4_PAD1	99	8	5	2	I/O	ST/CMOS	
SERCOM4_PAD2	16	—	—	1	I/O	ST/CMOS	
SERCOM4_PAD2	89	11	8	2	I/O	ST/CMOS	
SERCOM4_PAD3	2	—	—	2	I/O	ST/CMOS	
SERCOM4_PAD3	17	12	9	1	I/O	ST/CMOS	
SERCOM5_PAD0	11	—	—	2	I/O	ST/CMOS	SERCOM 5 I/O Pads
SERCOM5_PAD0	39	26	21	1	I/O	ST/CMOS	
SERCOM5_PAD1	12	—	—	2	I/O	ST/CMOS	
SERCOM5_PAD1	40	27	22	1	I/O	ST/CMOS	
SERCOM5_PAD2	13	—	—	2	I/O	ST/CMOS	
SERCOM5_PAD2	41	28	23	1	I/O	ST/CMOS	
SERCOM5_PAD3	4	—	—	2	I/O	ST/CMOS	
SERCOM5_PAD3	44	31	24	1	I/O	ST/CMOS	

Table 6-12. TCC Signal Descriptions

PIN NAME	PIN NUMBER			I/O Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN				
TCC0_WO0	25	—	—	2	I/O	ST/CMOS	TCC0 Waveform Outputs /Input Counter/ Capture
TCC0_WO0	35	—	—	3	I/O	ST/CMOS	
TCC0_WO0	51	33	25	1	I/O	ST/CMOS	
TCC0_WO1	24	—	—	2	I/O	ST/CMOS	
TCC0_WO1	36	—	—	3	I/O	ST/CMOS	
TCC0_WO1	54	34	26	1	I/O	ST/CMOS	
TCC1_WO0	58	37	27	1	I/O	ST/CMOS	TCC1 Waveform Outputs /Input Counter/ Capture
TCC1_WO0	63	39	—	2	I/O	ST/CMOS	
TCC1_WO1	61	38	28	1	I/O	ST/CMOS	
TCC1_WO1	66	41	—	2	I/O	ST/CMOS	
TCC2_WO0	43	30	29	2	I/O	ST/CMOS	TCC2 Waveform Outputs /Input Counter/ Capture
TCC2_WO0	64	40	—	1	I/O	ST/CMOS	
TCC2_WO1	42	29	33	2	I/O	ST/CMOS	
TCC2_WO1	72	45	—	1	I/O	ST/CMOS	

Table 6-12. TCC Signal Descriptions (continued)

PIN NAME	PIN NUMBER			I/O Set	PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/ VQFN	48 TQFP/ VQFN				
TCC3_WO0	37	49	37	2	I/O	ST/CMOS	TCC3 Waveform Outputs /Input Counter/ Capture
TCC3_WO0	76	—	—	1	I/O	ST/CMOS	
TCC3_WO1	38	52	38	2	I/O	ST/CMOS	
TCC3_WO1	80	—	—	1	I/O	ST/CMOS	
TCC4_WO0	8	4	44	2	I/O	ST/CMOS	TCC4 Waveform Outputs /Input Counter/ Capture
TCC4_WO0	94	60	—	1	I/O	ST/CMOS	
TCC4_WO1	9	5	47	2	I/O	ST/CMOS	
TCC4_WO1	98	63	—	1	I/O	ST/CMOS	
TCC5_WO0	77	50	48	2	I/O	ST/CMOS	TCC5 Waveform Outputs /Input Counter/ Capture
TCC5_WO0	100	64	—	1	I/O	ST/CMOS	
TCC5_WO1	1	1	1	1	I/O	ST/CMOS	
TCC5_WO1	79	51	—	2	I/O	ST/CMOS	
TCC6_WO0	3	2	2	1	I/O	ST/CMOS	TCC6 Waveform Outputs /Input Counter/ Capture
TCC6_WO0	88	55	—	2	I/O	ST/CMOS	
TCC6_WO1	5	3	3	1	I/O	ST/CMOS	
TCC6_WO1	90	56	—	2	I/O	ST/CMOS	

Table 6-13. USB Signal Descriptions

PIN NAME	PIN NUMBER			PIN TYPE	FUNCTION TYPE	DESCRIPTION
	100 TQFP	64 TQFP/VQFN	48 TQFP/VQFN			
USB_SOF_1KHZ	80	52	38	O	CMOS	USB FS Start of Frame
USB_DM	83	53	39	I/O	Analog	USB FS D- differential data line
USB_DP	84	54	40	I/O	Analog	USB FS D+ differential data line

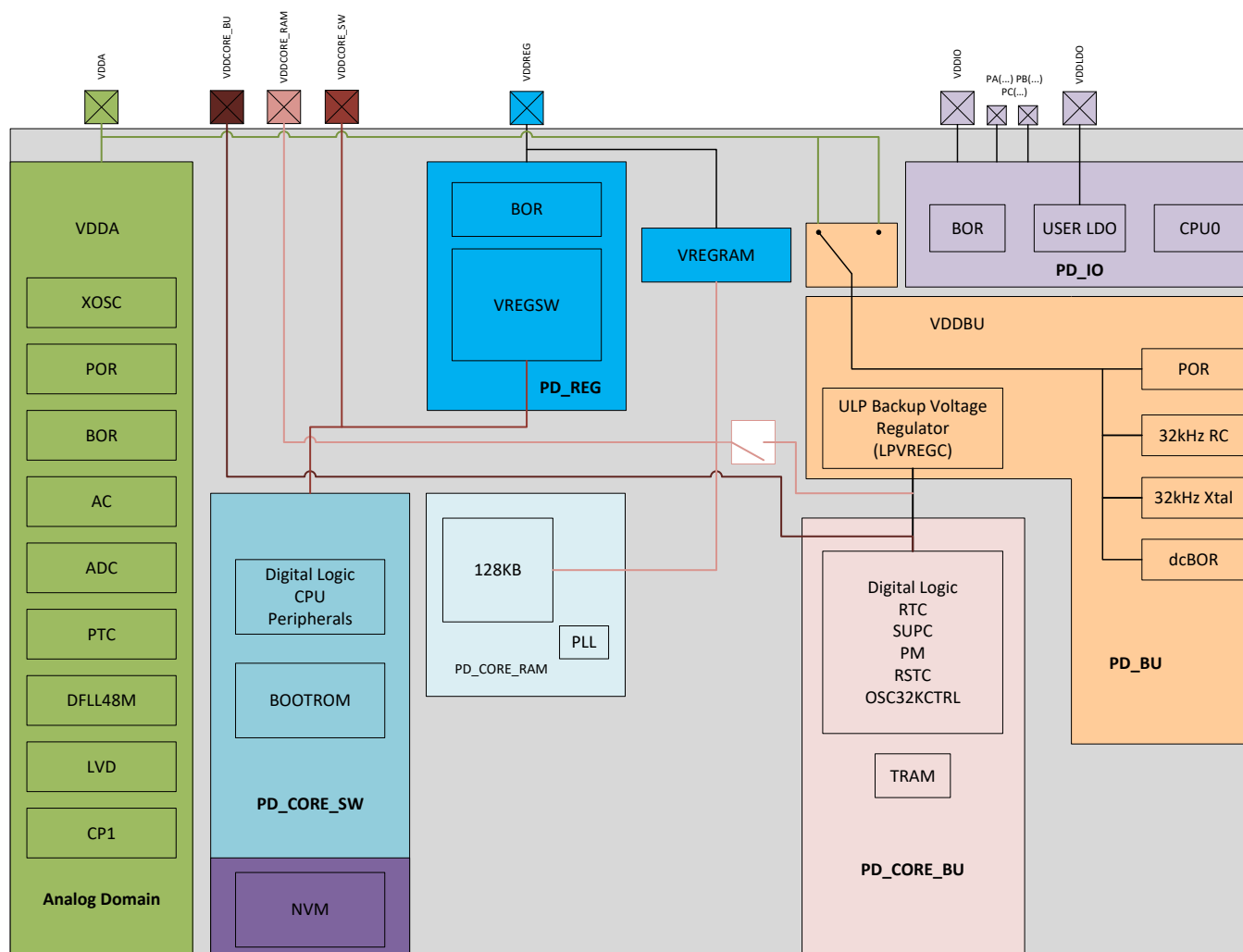
7. Power Supplies and Startup Considerations

Table 7-1. Power Supply Pins

Name	Associated Ground	Powers
VDDREG	GND	Voltage regulators for RAM, DFLL, Core, Peripherals (Nominal range: 1.71 V - 1.89 V, Recommended: 1.8 V)
VDDIO	GND	I/O Lines, XOSC, DFLL48 Flash Panels (by double bonds to VDD Flash pins) Charge Pumps for ADC, AC, and PTC (Nominal range: 1.63V-3.63V, Recommended: 3.3 V)
AVDD	AVSS	Analog Peripherals ADC, AC, and PTC POR and BORs, BGAP (Band Gap Voltage Reference) Backup Power Domain (XOSC32K, FRC32K) VDDCORE Backup Power Domain (RTC, OSC32KCTRL, TRAM, SUPC, PM, RSTC, HSM RTC and Tamper) (Nominal range: 1.63 V-3.63 V, Recommended: 3.3 V)

7.1 Power Domain Overview

Figure 7-1. Power Domain Block Diagram



7.2 Power Domain Constraints

The PIC32CM SG00/GC00 power domains are not independent of each other:

- VDDCORE, VDDIO and VDDREG share GND, whereas AVDD has its own ground, AVSS

- AVDD and VDDIO must share the main supply, VDD
- VDD_BU, VDDREG, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW, AVDD, VDDIO, and VDDFLASH are internal power domains

The maximum supply falling and rising rates of the different power supplies must not exceed the values described in the “Power Supply Characteristics” section of the [Electrical Characteristics](#) chapter.

7.2.1 Power Supplies

The PIC32CM SG00/GC00 has the following power supply pins:

- AVDD powers the Backup power domain as well as the ADC, Analog Comparator (AC), and the Peripheral Touch Controller (PTC)
- VDDREG powers the internal regulators for the VDDCORE_SW, VDDCORE_RAM, and VDDCORE_BU power domains
- VDDIO powers I/O lines, an External Crystal Oscillator (XOSC), the 48 MHz Digital Frequency Locked Loop (DFLL48), and two charge pumps which support the Analog Comparator (AC), 48 MHz Oscillator, and the ADCs on the device (ADC0 and PTC ADC). Most VDDIO pins also power the device's Flash panel through double bonding.

For decoupling recommendations for the different power supplies, refer to the schematic checklist.

7.2.2 Power-on Reset (POR) and Brown-out Detectors (BOD)

The PIC32CM SG00/GC00 embeds three features to monitor, warn, and reset the device:

- A Power-on Reset on V_{DD} (AVDD and VDDIO):
 - Monitoring is always activated, including during device startup or during any sleep modes
 - Having V_{DD} below a fixed threshold voltage will reset the whole device

Note: Refer to the [Electrical Characteristics](#) for the rising and falling threshold voltages.

- A Brown-out Detector (VddbOD) on V_{DD} (AVDD and VDDIO):
 - The VddbOD monitors VDD continuously
 - dcBOR in (continuous mode) or periodically (sampled mode) with a programmable sample frequency in Backup mode
 - A programmable threshold loaded from the NVM User Row is used to trigger an interrupt and/or reset the whole device

- PORCORE a Brown-out Detector on VDDCORE

Note: PORCORE is calibrated in production and its calibration parameters are stored in the NVM User Row. These data must not be changed to ensure correct device behavior.

7.2.3 Voltage Regulator

The PIC32CM SG00/GC00 internal Main Voltage Regulators have three different modes, controlled by the Power Manager (PM).

- **Enabled mode:** This is the default mode when CPU and peripherals are running.
- **Low-Power mode:** This is the mode when CPU and peripherals are in standby.
- **Disabled mode:** When the chip is in backup mode, the internal regulator is off, the VDDCORE_SW core power domain is OFF. The VDDCORE_BU backup domain is powered by the backup regulator (LPVREGC).

7.2.4 Internal Power Domains

There are a variety of internal regulated power domains (nominal 1.2v) as shown:

- **VDDCORE_BU:** Powers the backup domain

- **VDDCORE_SW**: This is the main voltage domain for the CPU, bus and most peripherals
- **VDDCORE_RAM**: This domain is used to retain the System SRAM

7.3 Power Up

This section summarizes the power-up sequence of the PIC32CM SG00/GC00. The behavior after power-up is controlled by the Power Manager (RPMU /PM).

7.3.1 Starting of Internal Regulator

After power-up of the VDDREG, VDDIO and AVDD supplies ($VDDx > \text{POR voltage}$), the device is set to its initial state and kept in Reset, until the band-gap and internal regulator power has stabilized throughout the device.

The internal regulator provides the internal VDDCORE_SW once the external voltage VDDIO/AVDD and the internal VDDCORE_xx domains reach a stable value, the internal System Reset is released.

7.3.2 Starting of Clocks

Once the power has stabilized and the internal Reset is released, the device will use a 48 MHz clock by default. The clock source for this clock signal is DFLL48M, which is enabled after a reset by default. This is also the default time base for Generic Clock Generator 0. In turn, Generator 0 provides the main clock GCLK_MAIN which is used by the Main Clock module (MCLK).

Some synchronous system clocks are active after Start-Up, allowing software execution. Refer to the *Clock Mask Registers* section in the *Main Clock (MCLK)* document for the list of clocks that are running by default. Synchronous system clocks that are running receive the 48 MHz clock from Generic Clock Generator 0. Other generic clocks are disabled.

7.3.3 I/O Pins

After power-up, the I/O pins are tri-stated except $\overline{\text{RESET}}$, which is pull-up enabled and configured as an input.

7.3.4 Fetching of Initial Instructions

After Reset has been released, the CPU starts fetching PC and SP values from the Reset address, 0x04000000. This points to the first executable address in the internal Flash memory. The code read from the internal Flash can be used to configure the clock system and clock sources. See the related peripheral documentation for details. Refer to the Arm "Architecture Reference Manual" for more information on CPU startup (www.arm.com).

7.4 Analog Peripherals Considerations

This chapter provides a global view of the analog system, which is composed of the following analog peripherals: AC and ADC.

The analog peripherals can be connected to each other as illustrated in the block diagrams for the individual analog modules.



Important:

When an analog peripheral is enabled, each analog output of the peripheral will be prevented from using the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternate pad functions.

7.5 Device Startup

This section summarizes the PIC32CM SG00/GC00 device startup sequence which starts after device power-up. After power-up, the device is kept in reset until the power has stabilized throughout

the device. Once VDDIO/AVDD and VDDCORE voltages reach a stable value, the internal reset is released.

7.5.1 Clocks Startup

The device selects the OSC48M oscillator which is enabled by default after reset and configured at 48 MHz.

This 48 MHz clock is also the default time base for the Generic Clock Generator 0 which provides the main clock (CLK_MAIN) to the system through the GLCK_MAIN clock.

Note: Other generic clocks are disabled to optimize power consumption.

Some synchronous clocks require also to be active after startup.

Note: These active synchronous clocks also receive the 48 MHz clock from Generic Clock Generator 0.

Refer to the *Clock Mask Register* section in the *Main Clock (MCLK)* chapter to obtain the list of clocks that are running by default.

7.5.2 I/O Pins

After reset, the I/O pins are tri-stated except PC08 pin (configured as an input with pull-up enabled) which is by default assigned to the SWCLK peripheral function to allow debugger probe detection.

8. Product Mapping

Table 8-1. Address Space

Start Address	End Address	Size (KB)	Name
0x04000000	0x0400FFFF	64	ROM
0x04010000	0x07FFFFFF	–	Reserved
0x08000000	0x08003FFF	16	BFM
0x08004000	0x09FFFFFF	–	Reserved
0x0A000000	0x0A007FFF	32	CFM
0x0A008000	0x0BFFFFFF	–	Reserved
0x0C000000	0x0C07FFFF	512	PFM
0x0C080000	0x1FFFFFFF	–	Reserved
0x20000000	0x2001FFFF	128	RET
0x20020000	0x43FFFFFF	–	Reserved
0x44000000	0x44033FFF	208	APB0
0x44034000	0x447FFFFFFF	–	Reserved
0x44800000	0x44839FFF	232	APB1
0x4483A000	0x44FFFFFF	–	Reserved
0x45000000	0x4500FFFF	64	APB2
0x45010000	0x4EFFFFFF	–	Reserved
0x4F000000	0x4F017FFF	96	HSM

Table 8-2. CFM

Start Address	End Address	Size (KB)	Name
0x0A000000	0x0A000FFF	4	BOOTCFG1A
0x0A001000	0x0A001FFF	4	USEROTP
0x0A002000	0x0A002FFF	4	BOOTCFG1
0x0A003000	0x0A003FFF	4	ROMCFG
0x0A004000	0x0A004FFF	4	VSS0
0x0A005000	0x0A005FFF	4	VSS1
0x0A006000	0x0A006FFF	–	Reserved
0x0A007000	0x0A007FFF	4	CalOTP

Table 8-3. APB0

Start Address	End Address	Size (KB)	Name
0x44000000	0x44001FFF	8	DSU
0x44002000	0x44003FFF	8	FCR
0x44004000	0x44005FFF	8	FCW
0x44006000	0x44007FFF	8	OSCCTRL
0x44008000	0x44009FFF	8	GCLK
0x4400A000	0x4400BFFF	8	MCLK
0x4400C000	0x4400DFFF	8	FREQM
0x4400E000	0x4400FFFF	8	WDT
0x44010000	0x44011FFF	8	HMATRIX2
0x44012000	0x44013FFF	8	IDAU
0x44014000	0x4401FFFF	–	Reserved
0x44020000	0x44021FFF	8	EIC

Table 8-3. APB0 (continued)

Start Address	End Address	Size (KB)	Name
0x44022000	0x44023FFF	8	PAC
0x44024000	0x44025FFF	–	Reserved
0x44026000	0x44027FFF	–	Reserved
0x44028000	0x44029FFF	–	Reserved
0x4402A000	0x4402BFFF	–	Reserved
0x4402C000	0x4402DFFF	8	MCRAMC
0x4402E000	0x4402FFFF	8	CAN0
0x44030000	0x44031FFF	8	CAN1
0x44032000	0x44033FFF	8	H2PB0

Table 8-4. APB1

Start Address	End Address	Size (KB)	Name
0x44800000	0x44801FFF	8	PORT
0x44802000	0x44803FFF	8	DMAC
0x44804000	0x44805FFF	8	BROMC
0x44806000	0x44807FFF	8	EVSYS
0x44808000	0x44809FFF	8	SERCOM0
0x4480A000	0x4480BFFF	8	SERCOM1
0x4480C000	0x4480DFFF	8	SERCOM2
0x4480E000	0x4480FFFF	8	SERCOM3
0x44810000	0x44811FFF	8	TCC0
0x44812000	0x44813FFF	8	TCC1
0x44814000	0x44815FFF	8	TCC2
0x44816000	0x44817FFF	8	TCC3
0x44818000	0x44819FFF	8	ADC
0x4481A000	0x4481FFFF	–	Reserved
0x44820000	0x44821FFF	8	SERCOM4
0x44822000	0x44823FFF	8	SERCOM5
0x44824000	0x44825FFF	8	TCC4
0x44826000	0x44827FFF	8	TCC5
0x44828000	0x44829FFF	8	TCC6
0x4482A000	0x4482BFFF	8	AC
0x4482C000	0x4482DFFF	8	PTC
0x4482E000	0x4482FFFF	8	CCL0
0x44830000	0x44831FFF	8	CCL1
0x44832000	0x44833FFF	8	USB
0x44834000	0x44835FFF	8	PUF
0x44836000	0x44837FFF	–	Reserved
0x44838000	0x44839FFF	8	H2PB1

Table 8-5. APB2

Start Address	End Address	Size (KB)	Name
0x45000000	0x45001FFF	8	PM
0x45002000	0x45003FFF	8	SUPC
0x45004000	0x45005FFF	8	RSTC
0x45006000	0x45007FFF	8	RTC

Table 8-5. APB2 (continued)

Start Address	End Address	Size (KB)	Name
0x45008000	0x45009FFF	8	OSC32KCTRL
0x4500A000	0x4500BFFF	8	TRAM
0x4500C000	0x4500DFFF	8	AT
0x4500E000	0x4500FFFF	8	H2PB2

9. Peripherals

9.1 Register Description

9.1.1 Registers Properties

Registers can be 8, 16, or 32 bits wide. Atomic 8-bit, 16-bit and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

PAC Write-Protection Register Property:

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). The PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For more details, refer to the *PAC - Peripheral Access Controller*.

Read-Synchronized, Write-Synchronized Register Property:

Some registers (or bit fields within a register) require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" (bits) and "Write-Synchronized" (bits) property in each individual register description. For more details, refer to *Register Synchronization*.

Enable-Protected Register Property:

Some registers (or bit fields within a register) can only be written when the peripheral is disabled. Such protection is denoted by the "Enable-Protected" (bits) property in each individual register description.

Mix-Secure Peripherals Register Property:

A Mix-Secure Peripheral has different types of registers (Non-Secure, Secure, Write-Secure, and Mix-Secure) with different access permissions for each bit field. Refer to the PORT and EIC Specific Secure Access Protection chapters.

In the following register descriptions, the access permissions are specified.

For more details, refer to *Peripherals Security Attribution*.

Bitfield Access Properties:

The access properties of bit fields within a register are defined as follows:

Table 9-1. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

9.1.2 Registers Access Permissions

For PIC32CM GC devices, there is no notion of secure vs non-secure: all peripherals are accessible. The rest of this chapter is dedicated to PIC32CM SG devices.

For PIC32CM SG devices, peripherals are defined as secure or non-secure by the user through the BOOTCFG1/1A H2PB fuses, which are copied by the ROM code in the related registers, or by directly writing in the H2PB registers. The PORT and EIC, when defined as secure in H2PB will behave as Mix-Secure peripherals (See below). The HSM-Lite is always Secure. An empty or invalid BOOTCFG1/1A configuration will be replaced by a hardcoded configuration applied by the ROM code.

Each Peripheral has different register access permissions depending on its Security Attribution :

For peripherals configured as Secure:

- Non-Secure access is discarded and a synchronous bus error is generated
- Secure access is granted

For peripherals configured as Non-Secure:

- Secure and Non-Secure accesses are granted

For Mix-Secure peripherals (PORT and EIC):

- If the peripheral is configured as Non-Secure:
 - Secure access and Non-Secure access are granted
- If the peripheral is configured as Secure in the H2PB registers:
 - Refer to the POR and EIC chapters for details

9.1.3 Register Synchronization

9.1.3.1 Overview

Most of the peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

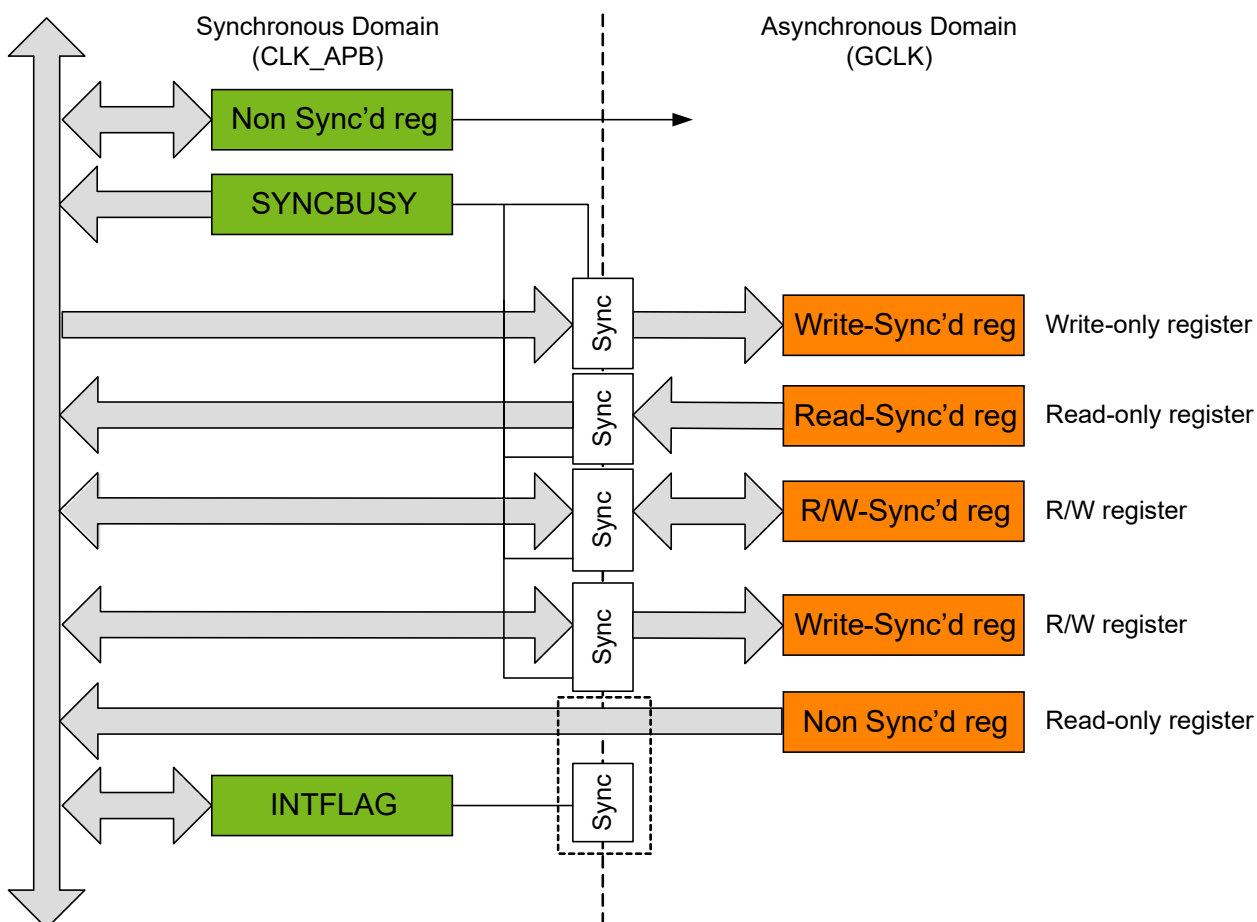
Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

As shown in the following figure, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).

Note: For registers requiring read-synchronization and write-synchronization, the corresponding bit in the SYNCBUSY register is shared.

Synchronization is denoted by the "Read-Synchronized" (bits) and "Write-Synchronized" (bits) property in each individual register description.

Figure 9-1. Register Synchronization Overview



9.1.3.2 General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain (GCLK). The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. For additional information, refer to the *Synchronization Delay*.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported through the Peripheral Access Controller (PAC).

For example, the REGA, REGB are 8-bit core registers, and the REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, hence multiple registers can be synchronized in parallel. Consequently, after the REGA (8-bit access) was written, the REGB (8-bit access) can be written immediately without error.

The REGC (16-bit access) can be written without affecting the REGA or REGB registers. If the REGC register is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated through the PAC.

A 32-bit access to offset 0x00 will write all three registers. The REGA, REGB, and REGC registers can be updated at different times because of independent write synchronization.

9.1.3.3 General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.

Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

9.1.3.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronization Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

9.1.3.5 Synchronization Delay

The synchronization will delay read and writer accesses by a certain amount. This delay D is within the range of:

$$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$$

Where,

P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

9.1.4 Interrupt Handling

As a general rule, interrupt flags must be cleared and then read back to confirm they are cleared before exiting the interrupt routine to avoid double interrupts.

9.1.5 Peripheral Dependencies

Table 9-2. Peripherals Configuration Summary

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL. PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB. TRIGSRC)	H2PBx. NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn. EVGEN)	Users (USERm)	Path Type			
H2PB0 Bridge Peripherals												
DSU	0x44000000		CLMSK0.0 (AHB)	0		0				1: DCC0	0.0	CORE_SW
										2: DCC1		
FCR	0x44002000	0: FCR	CLMSK0.1 (AHB)	0		1					0.1	CORE_SW
FCW	0x44004000	1: FCW	CLMSK0.2 (AHB)	0		2					0.2	CORE_SW
OSCCTRL	0x44006000	4: XOSCRDY	CLKMSK0.6 (APB)	0	0: DFLL48M	6	2: XOSCFAIL				0.3	CORE_SW
		4: XOSCFAIL			1: PLL							
		4: CLKFAIL										
		5: DFLLRDY										
		5: DFLLLOCK										
		5: DFLLLOVF										
		5: DFLLUNF										
		5: DFLLRCS										
		5: DFLLFAIL										
		6: PLLLOCKR_0										
6: PLLLOCKF_0												
GCLK	0x44008000		CLKMSK0.8 (APB)	0		8					0.4	CORE_SW
MCLK	0x4400A000	9: CKRDY	CLKMSK0.9 (APB)	0		9					0.5	CORE_SW
FREQM	0x4400C000	10: DONE	CLKMSK0.10 (APB)	0	2: MSR0	10	4: DONE	0: START	R		0.6	CORE_SW
		10: WINMON			3: MSR1		5: WINMON					
					4: REF							
WDT	0x4400E000	11: EW_A	CLKMSK0.11 (APB)	0		11					0.7	CORE_SW

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx. NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
HMATRIX2	0x44010000	47: HMATRIX	CLKMSK0.30 (AHB)	0		22					0.8	CORE_SW
			CLKMSK0.31 (APB)	0								
IDAU	0x44012000		CLKMSK1.2 (APB)	0		24					0.9	CORE_SW

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
EIC	0x44020000	16: EXTINT0	CLKMSK0.13 (APB)	0	5: EIC	13	21: EXTINT0				0.16	CORE_SW
		17: EXTINT1					22: EXTINT1					
		18: EXTINT2					23: EXTINT2					
		19: EXTINT3					24: EXTINT3					
		20: EXTINT4					25: EXTINT4					
		21: EXTINT5					26: EXTINT5					
		22: EXTINT6					27: EXTINT6					
		23: EXTINT7					28: EXTINT7					
		24: EXTINT8					29: EXTINT8					
		25: EXTINT9					30: EXTINT9					
		26: EXTINT10					31: EXTINT10					
		27: EXTINT11					32: EXTINT11					
		28: EXTINT12					33: EXTINT12					
		29: EXTINT13					34: EXTINT13					
		30: EXTINT14					35: EXTINT14					
		31: EXTINT15					36: EXTINT15					
		32: NSCHK										
PAC	0x44022000		CLKMSK0.14 (APB)	0		14					0.17	CORE_SW

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
MBISTINTF	0x44028000		CLKMSK0.18 (APB)	0							0.20	CORE_SW
			DCLK: 0.19	2								
MCRAMC	0x4402C000	34: INT				16					0.22	CORE_SW
CAN0	0x4402E000	35: INT0	CLKMSK0.23 (AHB)	0	6: CAN0	17					0.23	CORE_SW
		36: INT1										
		37: BERR										
		37: DBG										
CAN1	0x44030000	38: INT0	CLKMSK0.24 (AHB)	0	7: CAN1	18					0.24	CORE_SW
		39: INT1										
		40: BERR										
		40: DBG										
H2PB0	0x44032000		CLKMSK0.25 (AHB)	0		19					0.25	CORE_SW
			CLKMSK0.26 (APB)	0								
H2PB1 Bridge Peripherals												
PORT	0x44800000	41: NSCHK0	CLKMSK0.27 (APB)	0		20		2: EV0	R		1.0	CORE_SW
		41: NSCHK1						3: EV1	R			
		41: NSCHK2						4: EV2	R			
		41: NSCHK3						5: EV3	R			

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
DMAC	0x44802000	42: TCMPL0, SUSP0, TERR0	CLKMSK0.28 (AHB)	0		21	37: CH0	6: CH0	R		1.1	CORE_SW
		43: TCMPL1, SUSP1, TERR1					40: CH3	9: CH3	R			
		44: TCMPL2, SUSP2, TERR2										
		45: TCMPL3, SUSP3, TERR3										
		46: TCMPL4, SUSP4, TERR4										
		46: TCMPL5, SUSP5, TERR5										
		46: TCMPL6, SUSP6, TERR6										
		46: TCMPL7, SUSP7, TERR7										
BROMC	0x44804000		CLKMSK1.0 (AHB)	0		23					1.2	CORE_SW
			CLKMSK1.1 (APB)	0								

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
EVSYS	0x44806000	48: EVD0	CLKMSK1.3 (APB)	0	8: CH0	25					1.3	CORE_SW
		48: OVR0			9: CH1							
		49: EVD1			10: CH2							
		49: OVR1			11: CH3							
		50: EVD2			12: CH4							
		50: OVR2			13: CH5							
		51: EVD3			14: CH6							
		51: OVR3			15: CH7							
		52: EVD4			16: CH8							
		52: OVR4			17: CH9							
		53: EVD5			18: CH10							
		53: OVR5			19: CH11							
		54: EVD6										
		54: OVR6										
		55: EVD7										
		55: OVR7										
		56: EVD8										
		56: OVR8										
		57: EVD9										
		57: OVR9										
		58: EVD10										
		58: OVR10										
		59: EVD11										
		59: OVR11										
SERCOM0	0x44808000	60: 6	CLKMSK1.4 (APB)	0	20: SLOW	26				4: RX	1.4	CORE_SW
		61: 5			21: CORE					5: TX		
		62: 0										
		63: 1										
		64: 2										
		65: 3										
		66: 4										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
SERCOM1	0x4480A000	67: 6	CLKMSK1.5 (APB)	0	20: SLOW	27				6: RX	1.5	CORE_SW
		68: 5			22: CORE					7: TX		
		69: 0										
		70: 1										
		71: 2										
		72: 3										
		73: 4										
SERCOM2	0x4480C000	74: 6	CLKMSK1.6 (APB)	0	20: SLOW	28				8: RX	1.6	CORE_SW
		75: 5			23: CORE					9: TX		
		76: 0										
		77: 1										
		78: 2										
		79: 3										
		80: 4										
SERCOM3	0x4480E000	81: 6	CLKMSK1.7 (APB)	0	20: SLOW	29				10: RX	1.7	CORE_SW
		82: 5			24: CORE					11: TX		
		83: 0										
		84: 1										
		85: 2										
		86: 3										
		87: 4										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
TCC0	0x44810000	88: DFS	CLKMSK1.8 (APB)	0	25: TCC01	30	41: OVF	10: EVO	R	12: OVF	1.8	CORE_SW
		88: ERR					42: TRG	11: EV1	R	13: MC0		
		88: FAULTA					43: CNT	12: MC0	R	14: MC1		
		88: FAULTB					44: MC0	13: MC1	R			
		88: FAULT0					45: MC1					
		88: FAULT1										
		88: OVF										
		88: UFS										
		89: CNT										
		89: TRG										
		90: MC0										
		91: MC1										
TCC1	0x44812000	92: DFS	CLKMSK1.9 (APB)	0	25: TCC01	31	46: OVF	14: EVO	R	15: OVF	1.9	CORE_SW
		92: ERR					47: TRG	15: EV1	R	16: MC0		
		92: FAULTA					48: CNT	16: MC0	R	17: MC1		
		92: FAULTB					49: MC0	17: MC1	R			
		92: FAULT0					50: MC1					
		92: FAULT1										
		92: OVF										
		92: UFS										
		93: CNT										
		93: TRG										
		94: MC0										
		95: MC1										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
TCC2	0x44814000	96: DFS	CLKMSK1.10 (APB)	0	26: TCC2	32	51: OVF	18: EVO	R	18: OVF	1.10	CORE_SW
		96: ERR					52: TRG	19: EV1	R	19: MC0		
		96: FAULTA					53: CNT	20: MC0	R	20: MC1		
		96: FAULTB					54: MC0	21: MC1	R			
		96: FAULT0					55: MC1					
		96: FAULT1										
		96: OVF										
		96: UFS										
		97: CNT										
		97: TRG										
		98: MC0										
		99: MC1										
TCC3	0x44816000	100: DFS	CLKMSK1.11 (APB)	0	27: TCC3	33	56: OVF	22: EVO	R	21: OVF	1.11	CORE_SW
		100: ERR					57: TRG	23: EV1	R	22: MC0		
		100: FAULTA					58: CNT	24: MC0	R	23: MC1		
		100: FAULTB					59: MC0	25: MC1	R			
		100: FAULT0					60: MC1					
		100: FAULT1										
		100: OVF										
		100: UFS										
		101: CNT										
		101: TRG										
		102: MC0										
		103: MC1										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
ADC	0x44818000	130: REQ0	CLKMSK1.17 (APB)	0	33: ADC	39	76: RESRDY	38: TRIG0	A	37: PFFRDY	1.12	CORE_SW
		131: REQ1					77: CMP	39: TRIG1	A			
								40: TRIG2	A			
								41: TRIG3	A			
								42: TRIG4	A			
								43: TRIG5	A			
								44: TRIG6	A			
								45: TRIG7	A			
								46: TRIG8	A			
								47: TRIG9	A			
								48: TRIG10	A			
SERCOM4	0x44820000	104: 6	CLKMSK1.12 (APB)	0	28: CORE	34				24: RX	1.16	CORE_SW
		105: 5								25: TX		
		106: 0										
		107: 1										
		108: 2										
		109: 3										
		110: 4										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
SERCOM5	0x44822000	111: 6	CLKMSK1.13 (APB)	0	29: CORE	35				26: RX	1.17	CORE_SW
		112: 5								27: TX		
		113: 0										
		114: 1										
		115: 2										
		116: 3										
		117: 4										
TCC4	0x44824000	118: DFS	CLKMSK1.14 (APB)	0	30: TCC4	36	61: OVF	26: EV0	R	28: OVF	1.18	CORE_SW
		118: ERR					62: TRG	27: EV1	R	29: MC0		
		118: FAULTA					63: CNT	28: MC0	R	30: MC1		
		118: FAULTB					64: MC0	29: MC1	R			
		118: FAULT0					65: MC1					
		118: FAULT1										
		118: OVF										
		118: UFS										
		119: CNT										
		119: TRG										
		120: MC0										
		121: MC1										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
TCC5	0x44826000	122: DFS	CLKMSK1.15 (APB)	0	31: TCC5	37	66: OVF	30: EV0	R	31: OVF	1.19	CORE_SW
		122: ERR					67: TRG	31: EV1	R	32: MC0		
		122: FAULTA					68: CNT	32: MC0	R	33: MC1		
		122: FAULTB					69: MC0	33: MC1	R			
		122: FAULT0					70: MC1					
		122: FAULT1										
		122: OVF										
		122: UFS										
		123: CNT										
		123: TRG										
		124: MC0										
		125: MC1										
TCC6	0x44828000	126: DFS	CLKMSK1.16 (APB)	0	32: TCC6	38	71: OVF	34: EV0	R	34: OVF	1.20	CORE_SW
		126: ERR					72: TRG	35: EV1	R	35: MC0		
		126: FAULTA					73: CNT	36: MC0	R	36: MC1		
		126: FAULTB					74: MC0	37: MC1	R			
		126: FAULT0					75: MC1					
		126: FAULT1										
		126: OVF										
		126: UFS										
		127: CNT										
		127: TRG										
		128: MC0										
		129: MC1										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
AC	0x4482A000	132: COMP0	CLKMSK1.18 (APB)	0	34: AC	40	78: COMP0	49: SOC0	R		1.21	CORE_SW
		132: COMP1					79: COMP1	50: SOC1	R			
		132: WIN0					80: WIN0					
PTC	0x4482C000	133: EOC	CLKMSK1.19 (APB)	0	35: PTC	41	81: EOC	51: DSEQR	A	38: EOC	1.22	CORE_SW
		133: WCOMP					82: WCOMP	52: STCON V	A	39: SEQ		
		133: ACCRDY								40: WCOMP		
CCL0	0x4482E000		CLKMSK1.20 (APB)	0	36: CCL0	42	83: LUTOUT0	53: LUTIN 0	R		1.23	CORE_SW
							84: LUTOUT1	54: LUTIN 1	R			
							85: LUTOUT2	55: LUTIN 2	R			
							86: LUTOUT3	56: LUTIN 3	R			
CCL1	0x44830000		CLKMSK1.21 (APB)	0	37: CCL1	43	87: LUTOUT0	57: LUTIN 0	R		1.24	CORE_SW
							88: LUTOUT1	58: LUTIN 1	R			
							89: LUTOUT2	59: LUTIN 2	R			
							90: LUTOUT3	60: LUTIN 3	R			

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
USB	0x44832000	134: EORSMDNRSM	CLKMSK1.22 (AHB)	0 0	38: USB	44					1.25	CORE_SW
		134: EORSTRST	CLKMSK1.23 (APB)									
		134: LPMDCONN										
		134: LPMSUSPDDIS C										
		134: MSOF										
		134: RAMACER										
		134: RXSTPTXSTP7..0										
		134: STALL0STALL7..0										
		134: STALL17..10										
		134: SUSPEND										
		134: TRFAIL0TRFAIL7..0										
		134: TRFAIL1PERR7..0										
		134: UPRSM										
		134: WAKEUP										
		135: SOFHSOF										
		136: TRCPT17..00										
PUF	0x44834000	140: PUF	CLKMSK1.31 (APB)	0		49				41: PUFDI	1.26	CORE_SW
										42: PUFDO		

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx. NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
H2PB1	0x44838000		CLKMSK1.25 (AHB)	0		46					1.28	CORE_SW
			CLKMSK1.26 (APB)	0								
H2PB2 Bridge Peripherals												
PM	0x45000000	2: SLEEPRDY	CLKMSK0.3 (APB)	1		3					2.0	CORE_SW
SUPC	0x45002000	3: LVDDDET	CLKMSK0.4 (APB)	1		4	1: LVDET				2.1	CORE_SW
		3: LVDRDY										
		3: ULDDORDY										
		3: ULDOOVHEAT										
RSTC	0x45004000		CLKMSK0.5 (APB)	1		5					2.2	CORE_SW
RTC	0x45006000	12: TAMPER	CLKMSK0.12 (APB)	1		12	6: PER0	1: TAMPER	A	3: TIMES TAMP	2.3	CORE_BU
		13: OVf					7: PER1					
		14: PER7..0					8: PER2					
		15: CMP3..0					9: PER3					
							10: PER4					
							11: PER5					
							12: PER6					
							13: PER7					
							14: CMP0					
							15: CMP1					
							16: CMP2					
							17: CMP3					
							18: TAMPER					
							19: OVf					
							20: PERD					
OSC32KCTRL	0x45008000	7: XOSC32KRDY	CLKMSK0.7 (APB)	1		7	3: XOSC32KFAIL				2.4	CORE_BU
		8: XOSC32KFAIL										

Table 9-2. Peripherals Configuration Summary (continued)

Peripheral name	Base address	NVIC IRQ Index	MCLK		GCLK Peripheral Channel Index (PCHCTRLx)	PAC Peripheral Identifier Index (WRCTRL, PERID)	Events (EVSYS)			DMA Trigger Source Index (CHCTRLB, TRIGSRC)	H2PBx, NONSECy index	Power Domain
			(CLKMSKx.y)	CLKDIVx			Generators (CHANNELn, EVGEN)	Users (USERm)	Path Type			
TRAM	0x4500A000	33: ERR 33: DRP	CLKMSK0.15 (APB)	1		15					2.5	CORE_BU
H2PB2	0x4500E000		CLKMSK1.27 (AHB)	0		47					2.7	CORE_SW
			CLKMSK1.28 (APB)	1								
HSM-Lite												
HSM-Lite	0x4F000000	139	CLKMSK1.30 (AHB)	0		48						CORE_SW

10. Processor and Architecture

10.1 Processor and Architecture

Refer to the Arm reference documents "*ARM Cortex-M33 Processor Data Sheet*" (ARM DUI 0644) and "*ARM Cortex-M33 Technical Reference Manual*", available for download at the following location: www.arm.com.

10.2 Arm Cortex-M23 Configuration

Table 10-1. Arm Cortex-M23 Configuration

Feature	Configuration
ARMv8-M security (Trust Zone)	Yes
Memory Protection (MPU)	Regions: 8 Non-Secure and 8 Secure
Number of interrupt priority levels	2
Number of breakpoints comparators	4
Number of watch point comparators	2

10.3 Nested Vector Interrupt Controller

The Nested Vectored Interrupt Controller (NVIC) provides the user interface to control the external and on-chip peripheral interrupts. The NVIC provides up to 240 interrupt sources (unused sources are reserved for future use) that can be programmed with up to 4 or 8 different priority levels. The NVIC is included as part of the CPU. For additional information, refer to the "Arm Core Technical Reference Manual" available for download at www.arm.com.

10.4 NVIC Interrupt Mapping

Table 10-2. NVIC Interrupt Mapping

Module	Interrupt Index : Source
EIC - External Interrupt Controller	NMI : NMI
FCR - Flash Read Controller	0 : FCR
FCW - Flash Write Controller	1 : FCW
PM - Power Manager	2 : SLEEPDY
SUPC - Supply Controller	3 : LVDDDET/LVDRDY/ULDORDY/ULDOOVHEAT
OSCCTRL - Oscillators Control	4 : XOSCRDY/XOSCFAIL/CLKFAIL 5 : DFLLRDY/DFLLLOCK/DFLLOVF/DFLLUNF/DFLLRCS/DFLLFAIL 6 : PLLLOCKR_0/PLLLOCKF_0
OSC32KCTRL - 32 kHz Oscillators Control	7 : XOSC32KRDY 8 : XOSC32KFAIL
MCLK - Main Clock Controller	9 : CKRDY
FREQM - Frequency Meter	10 : DONE/WINMON
WDT - Watchdog Timer	11 : EW_A
RTC - Real-Time Counter	12 : TAMPER 13 : OVF 14 : PER0/PER1/PER2/PER3/PER4/PER5/PER6/PER7 15 : CMP0/CMP1/CMP2/CMP3

Table 10-2. NVIC Interrupt Mapping (continued)

Module	Interrupt Index : Source
EIC - External Interrupt Controller	16 : EXTINT0
	17 : EXTINT1
	18 : EXTINT2
	19 : EXTINT3
	20 : EXTINT4
	21 : EXTINT5
	22 : EXTINT6
	23 : EXTINT7
	24 : EXTINT8
	25 : EXTINT9
	26 : EXTINT10
	27 : EXTINT11
	28 : EXTINT12
	29 : EXTINT13
	30 : EXTINT14
	31 : EXTINT15
	32 : NSCHK
TRAM - TrustRAM	33 : ERR/DRP
MCRAMC - Multi-Channel RAM Controller	34 : INT
CAN0 - Control Area Network	35 : INT0
	36 : INT1
	37 : BERR/DBG
CAN1 - Control Area Network	38 : INT0
	39 : INT1
	40 : BERR/DBG
PORT - Port Module	41 : NSCHK0/NSCHK1/NSCHK2/NSCHK3
DMAC - Direct Memory Access Controller	42 : TCMPLO/SUSP0/TERR0
	43 : TCMP1/SUSP1/TERR1
	44 : TCMP2/SUSP2/TERR2
	45 : TCMP3/SUSP3/TERR3
	46 : TCMP4/SUSP4/TERR4/TCMP5/SUSP5/TERR5/TCMP6/SUSP6/TERR6/TCMP7/SUSP7/TERR7
HMATRIX2 - HSB Matrix	47 : HMATRIX
EVSYS - Event System Interface	48 : EVD0/OVR0
	49 : EVD1/OVR1
	50 : EVD2/OVR2
	51 : EVD3/OVR3
	52 : EVD4/OVR4
	53 : EVD5/OVR5
	54 : EVD6/OVR6
	55 : EVD7/OVR7
	56 : EVD8/OVR8
	57 : EVD9/OVR9
	58 : EVD10/OVR10
	59 : EVD11/OVR11

Table 10-2. NVIC Interrupt Mapping (continued)

Module	Interrupt Index : Source
SERCOM0 - Serial Communication Interface	60 : ERROR 61 : RXBRK 62 : DRE/PREC/MB 63 : TXC/AMATCH/SB 64 : RXC/DRDY 65 : RXS/SSL/TXFE 66 : CTSIC/RXFF
SERCOM1 - Serial Communication Interface	67 : ERROR 68 : RXBRK 69 : DRE/PREC/MB 70 : TXC/AMATCH/SB 71 : RXC/DRDY 72 : RXS/SSL/TXFE 73 : CTSIC/RXFF
SERCOM2 - Serial Communication Interface	74 : ERROR 75 : RXBRK 76 : DRE/PREC/MB 77 : TXC/AMATCH/SB 78 : RXC/DRDY 79 : RXS/SSL/TXFE 80 : CTSIC/RXFF
SERCOM3 - Serial Communication Interface	81 : ERROR 82 : RXBRK 83 : DRE/PREC/MB 84 : TXC/AMATCH/SB 85 : RXC/DRDY 86 : RXS/SSL/TXFE 87 : CTSIC/RXFF
TCC0 - Timer Counter for Control Applications	88 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 89 : CNT/TRG 90 : MC0 91 : MC1
TCC1 - Timer Counter for Control Applications	92 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 93 : CNT/TRG 94 : MC0 95 : MC1
TCC2 - Timer Counter for Control Applications	96 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 97 : CNT/TRG 98 : MC0 99 : MC1
TCC3 - Timer Counter for Control Applications	100 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 101 : CNT/TRG 102 : MC0 103 : MC1

Table 10-2. NVIC Interrupt Mapping (continued)

Module	Interrupt Index : Source
SERCOM4 - Serial Communication Interface	104 : ERROR 105 : RXBRK 106 : DRE/PREC/MB 107 : TXC/AMATCH/SB 108 : RXC/DRDY 109 : RXS/SSL/TXFE 110 : CTSIC/RXFF
SERCOM5 - Serial Communication Interface	111 : ERROR 112 : RXBRK 113 : DRE/PREC/MB 114 : TXC/AMATCH/SB 115 : RXC/DRDY 116 : RXS/SSL/TXFE 117 : CTSIC/RXFF
TCC4 - Timer Counter for Control Applications	118 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 119 : CNT/TRG 120 : MC0 121 : MC1
TCC5 - Timer Counter for Control Applications	122 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 123 : CNT/TRG 124 : MC0 125 : MC1
TCC6 - Timer Counter for Control Applications	126 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS 127 : CNT/TRG 128 : MC0 129 : MC1
ADC - ADC Controller	130 : REQ0 131 : REQ1
AC - Analog Comparator Controller	132 : COMP0/COMP1/WIN0
PTC - Peripheral Touch Controller	133 : EOC/WCOMP/ACRRDY
USB - Full-Speed Universal Serial Bus	134 : EORSMDNRSM/EORSTRST/LPMDCONN/LPMSUSPDDISC/MSOF/RAMACER/ RXSTPTXSTP0/RXSTPTXSTP1/RXSTPTXSTP2/RXSTPTXSTP3/RXSTPTXSTP4/ RXSTPTXSTP5/RXSTPTXSTP6/RXSTPTXSTP7/STALL0STALL0/STALL0STALL1/ STALL0STALL2/STALL0STALL3/STALL0STALL4/STALL0STALL5/STALL0STALL6/ STALL0STALL7/STALL10/STALL11/STALL12/STALL13/STALL14/STALL15/STALL16/ STALL17/SUSPEND/TRFAIL0TRFAIL0/TRFAIL0TRFAIL1/TRFAIL0TRFAIL2/ TRFAIL0TRFAIL3/TRFAIL0TRFAIL4/TRFAIL0TRFAIL5/TRFAIL0TRFAIL6/ TRFAIL0TRFAIL7/TRFAIL1PERR0/TRFAIL1PERR1/TRFAIL1PERR2/TRFAIL1PERR3/ TRFAIL1PERR4/TRFAIL1PERR5/TRFAIL1PERR6/TRFAIL1PERR7/UPRSM/WAKEUP 135 : SOFHSOF 136 : TRCPT00/TRCPT01/TRCPT02/TRCPT03/TRCPT04/TRCPT05/TRCPT06/TRCPT07 137 : TRCPT10/TRCPT11/TRCPT12/TRCPT13/TRCPT14/TRCPT15/TRCPT16/TRCPT17
AT - Anti-Tamper Controller (AT)	138 : TAMPER
HSM - Cryptographic Acceleration Module Lite	139 : CRYPTOMASTER/RNG/PUBKEY
PUF - Physically Unclonable Function	140 : PUF

10.5 High Speed Bus

The Bus Matrix implements a multi-layer AHB (Advanced High-performance Bus), based on the AHB-Lite protocol, that enables parallel access paths between multiple AHB hosts and clients in a system, therefore increasing the overall bandwidth. The Bus Matrix interconnects up to 16 AHB hosts to up to 16 AHB clients. The normal latency to connect a host to a client is one cycle except for the default host of the accessed client which is connected directly (zero cycle latency).

10.6 Bus Matrix Connectivity

Table 10-3. Connectivity Between Initiators and Targets

		Host #	Bus Clients								
			FCR	MCRAMC			BROMC	APB0	APB1	APB2	CAM
				AHB0	AHB1	AHB2					
Bus Hosts	Client		0	1	2	3	4	5	6	7	8
	CORE	0	x	x			x	x	x	x	x
	DMAC	1	x			x		x	x	x	
	CAN0	2			x						
	CAN1	3			x						
	FCW	4			x						
	CAM	5	x		x		x				

Note:

- Connectivity between a host and client is represented by the white colored cells in this table. If the cell is empty, there is no connectivity between the host and client. If there is connectivity the cell will contain a bullet "x".

11. Memories

11.1 Embedded Memories

The following are key features of the embedded memories:

- Flash memories are made of 4 KB pages
- 512 KB in-system self-programmable Flash (PFM) with Error Correction Code (ECC)
- 16 KB of Boot Flash Memory (BFM)
- 32 KB of Configuration Flash Memory (CFM)
 - 2 Variable Slot Storage (VSS0/1) pages
 - 1 User OTP page (USEROTP)
 - 1 ROM Configuration page (ROMCFG)
 - 1 Calibration OTP page (CALOTP)
 - 2 Boot Configuration pages (BOOTCFG1/1A)
- 128 KB SRAM with ECC

11.2 Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed and are never remapped in any way, even during boot. The 32-bit physical address space is mapped as described in the Product Mapping chapter.

11.3 SRAM Memory Configuration

Retention

In Standby mode, all the SRAM is retained. In Hibernate mode, only the upper 64 KB are retained (The lower 64 KB are used by the ROM code and wiped after usage).

RAM Error Correction

For safety applications, the PIC32CM SG00/GC00 family of devices embed SRAM error correction codes (ECC) to detect and correct single bit errors, or to enable dual-error detection for the system memory. The ECC feature can be enabled or disabled after reset through the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse.

11.4 Flash Memories

This device comes with a 512 KB Program Flash Memory (PFM), a 16 KB Boot Flash Memory (BFM), and a 32 KB Configuration Flash memory (CFM). They all support ECC for correcting on the fly ECC errors or detecting double errors.

11.4.1 Program and Boot Flash Memory (PFM and BFM)

Refer to the Chapter 31 *Non-Volatile Memory Controller* for details about Programming, Erasing, Reading these memories.

11.4.2 Configuration Flash Memory (CFM)

The CFM contains the following pages:

- **CALOTP:** This page contains read only calibration values. Some values need to be copied by the application software in ADC, PTC or USB before these peripherals can be used.
- **VSS0/1:** Two Variable Slot Storage for storing security assets.
- **ROMCFG:** This page contains:
 - The PUF Activation Code (AC)

- The Debug Access Level (DAL) value
- The tamper log
- The Boot ROM interactive mode command keys and locks
- Pre-boot configuration settings for running MBIST on Boot or Boot ROM CRC
- **BOOTCFG1/1A:** These pages contain the boot configuration selected by the user and applied by the ROM code. They must contain the same duplicate information. The ROM code will automatically switch from one to the other in case of errors found.
- **USEROTP:** 1 page dedicated for user application. It could be used to store a serial number, some user defined keys, or custom analog calibration values, and so on. Once written, each row can be locked independently forever, refer to FCW.UOWP for additional information. Once protected and locked, it cannot be erased using a page erase or chip erase.

11.4.3 Boot Configuration

The BOOTCFG1 and BOOTCFG1A boot configuration pages contain the following:

- Configuration settings
- Security settings
- Firmware metadata

The BOOTCFG1 and BOOTCFG1A pages share the same mapping and must be programmed identically. The redundant copy allows the copy to switch to the second page in case an error is found in the first one.

The BOOTCFG pages mapping is provided in the following table:

Table 11-1. BOOTCFG Page Mapping

Page	Address Range	Description
BOOTCFG1A	0x0A000000	Firmware Meta Data
	0x0A0007FF	
	0x0A000800	Registers
	0x0A0008D0	
BOOTCFG1	0x0A002000	Firmware Meta Data
	0x0A0027FF	
	0x0A002800	Registers
	0x0A0028D0	

11.4.4 ROMCFG Configuration

The DAL fuses in the ROMCFG page allows configuring the Debugger Access Level. Refer to the DSU for details.

All ROMCFG fuses are described in the following sections.

11.4.5 CAL-OTP Configuration

The following values are programmed at the factory and need to be copied by the application software in the associated peripheral registers.

Table 11-2. Factory Programmed Values to be Copied in Peripheral Registers

Address	Name	Description
0x0A007180	FCCFG64	Application software must copy FCCFG64[7:0] in PTC.CALIB
0x0A007184	FCCFG65	Application software must copy FCCFG65[31:0] in ADC.ADCCTRL
0x0A007190	FCCFG68	Application software must copy FCCFG68[15:0] in USB.PADCAL

11.5 Unique ID (UID)

The PIC32CM SG00/GC00 provides a 128 bit (UNIQID n , $n = 0,1,2,3$) which conforms to the version 4 UUID format. See the following table for the location in Flash memory. The unique ID is programmed onto the device at manufacture.

For more information on the standard, refer to the RFC4122 at tools.ietf.org/html/rfc4122.

Table 11-3. Unique ID Map

Address	Name	Description
0x0A0071E0	UNIQID0	UNIQID[31:0]
0x0A0071E4	UNIQID1	UNIQID[63:32]
0x0A0071E8	UNIQID2	UNIQID[95:64]
0x0A0071EC	UNIQID3	UNIQID[127:96]

12. PIC32CM SG00/GC00 Security Features

The following are security features of the PIC32CK SG/GC family of devices:

- Built-in cryptographic accelerator accessible through cryptographic libraries stored in ROM:
 - Supporting AES-128 encryption/decryption, SHA-256 authentication, GCM encryption, and authentication
 - Cryptographic libraries are especially designed for side channel and fault injection attacks prevention
- One True Random Generator (TRNG)
- Secure Boot, which performs integrity check on a configurable portion of the Flash (BS memory area)
- **Data Flash**
 - Optimized for secrets storage
 - Data Scrambling with user-defined key (optional)
 - Rapid Tamper erase on scrambling key and on one user-defined row
 - Silent access for side channel attack resistance
- TrustRAM

12.1 Data Flash

Refer to the *Non-Volatile Memory Controller (NVMCTRL)* chapter to get all security features related to the Data Flash.

12.2 TrustRAM

Refer to the *TrustRAM (TRAM)* chapter for all security features related to the TrustRAM.

12.3 Arm TrustZone for ARMv8-M

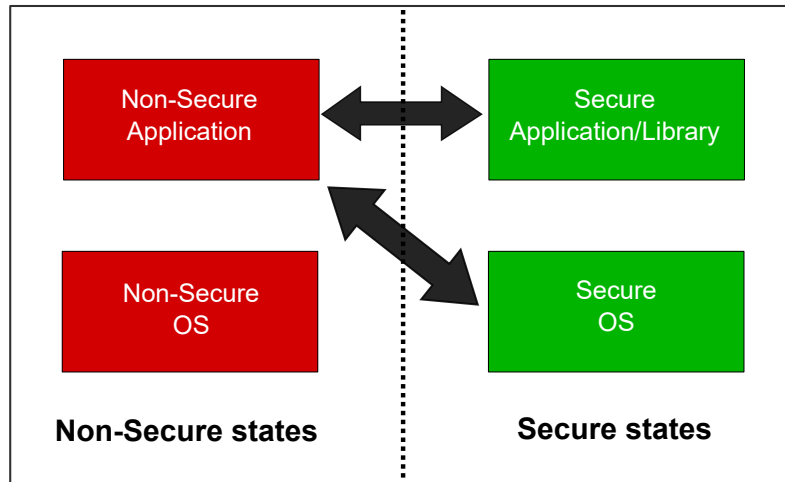
Arm TrustZone for ARMv8-M technology is an optional core extension implemented on PIC32CM SG/GC devices, which enables the system and the software to be partitioned into Secure and Non-Secure domains.

Secure software can access both Secure and Non-Secure memories and resources, while Non-Secure software can only access Non-Secure memories and resources.

Arm TrustZone for ARMv8-M allows Secure and Non-Secure code to run on a single CPU.

Note: The system always starts up in Secure state.

Figure 12-1. TrustZone for ARMv8-M



➔ Important: For additional information, refer to *TrustZone Technology for ARMv8-M Architecture*, which is available on the Arm web site (www.arm.com).

The memory space is partitioned into Non-Secure and Secure memory regions:

- **Non-Secure (NS):** Non-Secure addresses are used for memory and peripherals accessible by all software, that is, running on the device.
- **Secure (S):** Secure addresses are used for memory and peripherals accessible only by Secure software or Hosts.
- **Non-Secure Callable (NSC):** NSC is a special type of Secure memory location. It allows software to transition from Non-Secure to Secure state.

12.3.1 Peripherals Security Attribution

Security configurations related to the peripherals are handled by the PAC Controller.

Each peripheral can only be configured either in Secure or in Non-Secure mode except the IDAU peripheral which is always Secured.

The security configuration (Secure or Non-Secure) is propagated to each individual peripheral, thus it is the responsibility of the peripheral to grant or not the access with the following rules:

- If the peripheral is configured as Non-Secure in the PAC:
 - Secure and Non-Secure accesses are granted
- If the peripheral is configured as Secure in the PAC:
 - Secure access is granted
 - Non-Secure access is discarded (Write is ignored, read 0x0), a PAC error is triggered

 **Important:** These rules do not apply to the specific peripherals called Mix-Secure peripherals.

Note: The Secure application will usually provide an API for the Non-Secure application using the Non-Secure Callable region (NSC) to allow the Non-Secure application to request specific resources.

Table 12-1. Peripheral PAC Security Attribution (Excluding Mix-Secure Peripherals)

Mode	Secure Host Access	Non-Secure Host Access
Non-Secure	Read/Write	Read/Write
Secure	Read/Write	Discarded (Write ignored/Read 0x0) PAC Error is generated

 **Important:** Refer to “*Memory and Peripheral Security Configurations*” section for more details on the programming of NONSECA, NONSECB, and NONSECC parameters.

13. Implementation Defined Attribution Unit (IDAU)

13.1 Overview

The ARMv8-M Architecture defines a mechanism whereby an implementation defined mechanism (i.e., a component in the surrounding system) can define whether any particular address is exempt from checking, and is non-secure callable, or Secure. The Implementation Defined Attribution Unit partitions the memory map into IDAU regions which are subdivisions of the System Bus MAIN regions. The IDAU region size and decoding granularity are either fixed or configurable from the APB configuration registers. The MAIN region subdivisions are watermark based.

Note: This chapter only applies to the PIC32CMMSG variants.

13.2 IDAU Regions IDS

The Main region and associated IDAU Regions and Identifiers are show in the following table.

Table 13-1. Main and IDAU Regions Definition for PIC32CMMSG Devices

Start Address	End Address	Main Region	Main region type	IDAU region type	IDAU region identifier	valid region	Configurable size?
0x0400_0000	0x0400_FFFF	Boot rom	MR_SEC	SEC	1	1	-
0x0800_0000	0x0800_3FFF	BFM	MR_SEC_NSC_W	SEC	2	1	-
				NSC_W	3		Yes
0x0A00_0000	0x0A00_7FFF	CFM	MR_SEC	SEC	4	1	-
0x0C00_0000	0x0C07_FFFF	PFM	MR_SEC_NSC_NONSEC_W	SEC	5	1	-
				NSC_W	6		Yes
				NONSEC_W	7		Yes
0x2000_0000	0x2001_FFFF	SRAM	MR_SEC_NONSEC_W	SEC	8	1	-
				NONSEC_W	9		Yes
0x4400_0000	0x4403_FFFF	H2PB0	MR_EXEMPT	EXEMPT_B	10	1	-
0x4480_0000	0x4483_FFFF	H2PB1	MR_EXEMPT	EXEMPT_B	11	1	-
0x4500_0000	0x4500_FFFF	H2PB2	MR_EXEMPT	EXEMPT_B	12	1	-
0x4F00_0000	0x4F01_FFFF	HSMLITE	MR_SEC	SEC	13	1	-
Other	-	-	MR_INVALID	INVALID	0	0	-

13.3 IDAU Region Types

The IDAU region types are defined below:

Table 13-2. IDAU Region Types

IDAU REGION TYPE	DESCRIPTION
INVALID	Regions that are not decoded by the Bus matrix.
SEC	Secure region. Region which cannot be accessed by a Non-secure master. Non-secure transactions are discarded inside the CPU and a security fault is generated internally to the core.
NONSEC_W	Non-secure region. Region which is accessible by a non-secure transaction. When a Secure transaction targets a Non-secure region, transaction type is changed to Non-secure. (_W means that the region implements a watermark register).

Table 13-2. IDAU Region Types (continued)

IDAU REGION TYPE	DESCRIPTION
NSC_W	Non-secure callable region. Region which allows Non-secure software to call a Secure function, the first instruction in the called Secure function must be a SG (Secure Gateway). (_W means that the region implements a watermark register).
EXEMPT_B	Region exempt from attribution. The IDAU does not alter the transaction.



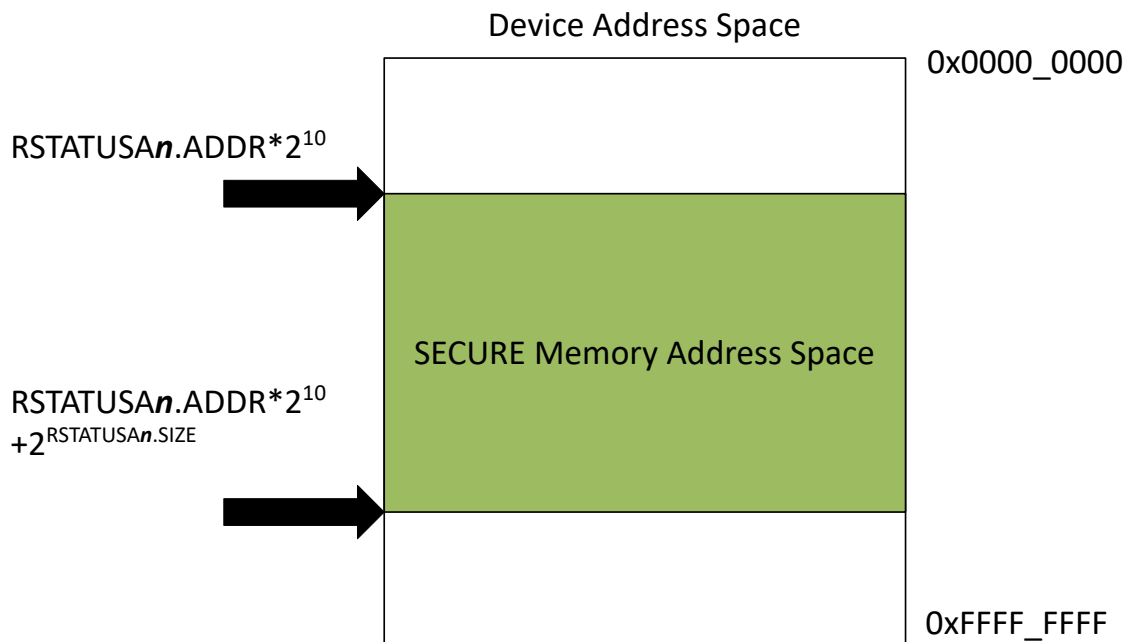
The _W and _B suffixes do not have the same meaning for the IDAU and MAIN regions.

13.4 Secure and Secure Always On Region Types

A secure region type is non-programmable region with a secure filtering attribute. When the CPU or other bus master requests access to an address range that falls within a secure region address space, the request is rejected with bus slave error if the bus request is non-secure. Secure requests are not rejected within the region address space. This region is only active if CTRLA.ENABLE=1. If CTRLA.ENABLE=0, this region is exempt from security filtering.

For this type of region, the RSTATUSA n .Type=0 and RSTATUSB n .NONSEC=0, where n is a value between from 1 to STATUS.NBRG. The lower address of the region is given by RSTATUSA n .ADDR[31:10] * 2^{10} and the upper address by RSTATUSA n .ADDR[31:10] * $2^{10} + 2^{RSTATUSA\mathbf{n}.SIZE} - 1$, when RSTATUSA n .SIZE is greater than 0. All addresses from the lower to the upper address range are subject to filtering. Attempting to program this type of region through RCTRL sets the STATUS.CFGERR and the command is ignored.

Figure 13-1. IDAU Secure Block Diagram



A secure always on region is similar to a secure region type except that it is always active regardless of CTRLA.ENABLE state.

13.5 IDAU Programmable Regions

Enter a short description of your topic here (optional).

A Programmable region is considered secure or non-secure (i.e. exempt) based on a user programmable security attribute. Bus requests to an address that falls within a programmable region are rejected with a slave bus error under the following condition

- request is non-secure (NONSEC=1) and the region is programmed as secure RSTATUSBn.NONSEC=0.

All other combinations are accepted.

For this type of region, the RSTATUSAn.Type=2 and RSTATUSBn.NONSEC reflects the security state of the region, where n is a value between from 1 to STATUS.NBRG. The lower address of the region is given by $RSTATUSAn.ADDR[31:10] * 2^{10}$ and the upper address by $RSTATUSAn.ADDR[31:10] * 2^{10} + 2^{RSTATUSAn.SIZE} - 1$, when RSTATUSAn.SIZE is greater than 0.

Program the security state through the RCTRL register by issuing the NONSEC or SEC command. This region is only active when CTRLA.ENABLE=1, otherwise it is treated as an exempt address range.

Figure 13-2. PROGRAMMABLE REGION TYPE WITH RSTATUSBn.NONSEC=0

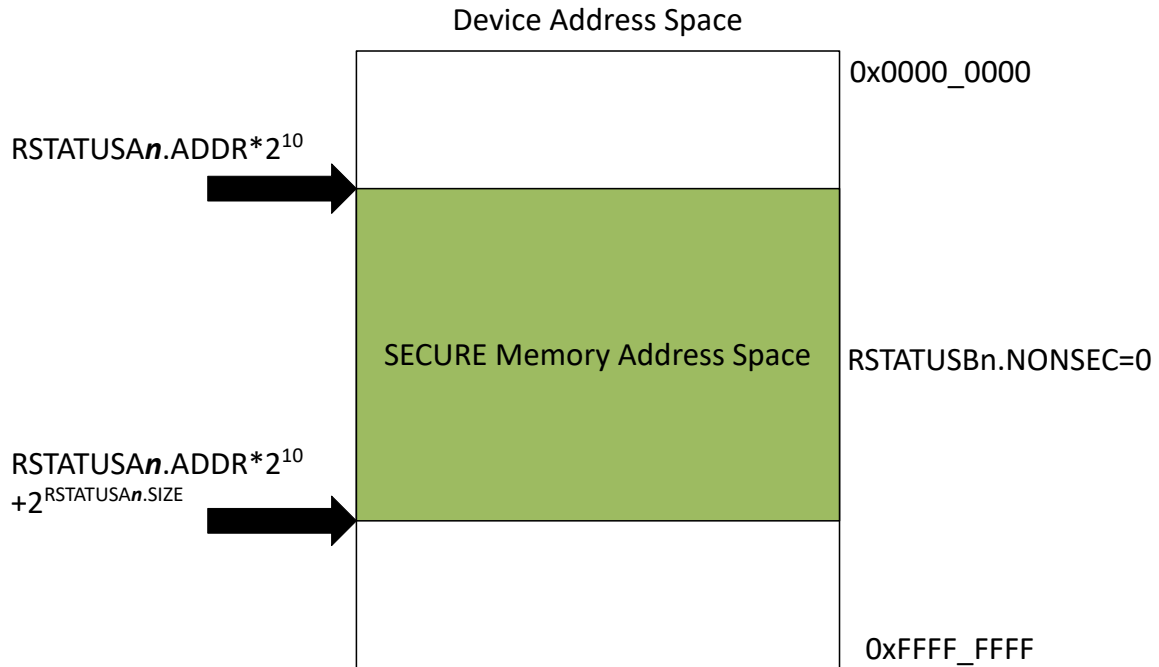
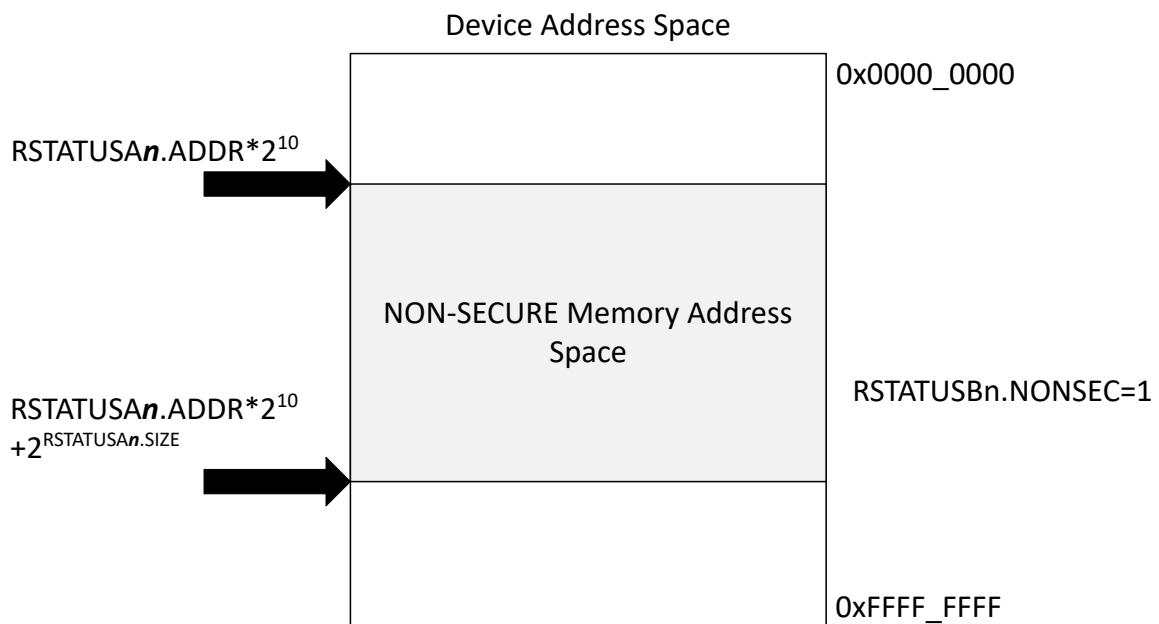


Figure 13-3. PROGRAMMABLE REGION TYPE WITH RSTATUSBn.NONSEC=1



13.6 IDAU Watermark Regions

Enter a short description of your topic here (optional).

Watermark regions are divided into a secure, non-secure callable, and non-secure subregions. The security attributes for each subregion is fixed but the size of the non-secure callable and non-secure

subregions are programmable. The size of the secure subregion is derived from programmed size of the non-secure callable and non-secure subregions, where the secure subregion size = region size - nonsecure callable subregion size - non-secure subregion size.

Security filtering is performed based on the subregion security attribute. Non-secure bus requests to an address within the secure subregion are rejected with a slave bus error. Non-secure data accesses to an address within the non-secure callable region are also rejected with a slave bus error, however, nonsecure instruction fetches are accepted. All other combinations are allowed.

For this type of region, the $RSTATUSA.n.TYPE=3$ and $2^{RSTATUSB.n.GRAN}$ specifies the granularity in bytes for subregion sizing.

The lower address of the secure subregion is given by $RSTATUSA.n.ADDR[31:10] * 2^{10}$ and the upper address by $RSTATUSA.n.ADDR[31:10] * 2^{10} + RSECSZ.n.SZM * 2^{RSTATUSB.n.GRAN} - 1$

The lower address of the non-secure callable subregion is given by $RSTATUSA.n.ADDR[31:10] * 2^{10} + RSECSZ.n.SZM * 2^{RSTATUSB.n.GRAN}$ and the upper address is $RSTATUSA.n.ADDR[31:10] * 2^{10} + (RSECSZ.n.SZM + RNSCSZ.n.SZM) * 2^{RSTATUSB.n.GRAN} - 1$ when $RNCSZ.n.SZM$ is greater than 0.

The lower address of the non-secure subregion is $RSTATUSA.n.ADDR[31:10] * 2^{10} + (RSECSZ.n.SZM + RNCSZ.n.SZM) * 2^{RSTATUSB.n.GRAN}$ and the upper address is given by $RSTATUSA.n.ADDR[31:10] * 2^{10} + 2^{RSTATUSA.n.SIZE} - 1$ when $RNSSZ.n.SZM$ is greater than 0.

$RSECSZ.n.SZM$ provides the size of the secure subregion, $RNCSZ.n.SZM$ the size of the non-secure callable subregion, and $RNSSZ.n.SZM$ the size of the non-secure subregion. The $RSECSZ.n.SZM$ field is updated each time the application updates the non-secure callable or non-secure subregions sizes. A subregion does not manifest if $R * SZ.n.SZM=0$.

Program the subregion sizes through the $RCTRL$ register by issuing the $NCSZ$ or $NSSZ$ command and providing a sizing value in $RCTRL.ARG$ equal to the desired size in bytes divided by $2^{RSTATUSB.n.GRAN}$. Attempting to size the non-secure callable or non-secure subregions greater than $2^{RSTATUSA.n.SIZE}$ or sizing either region such that the combined size of non-secure callable and non-secure subregions is greater than $2^{RSTATUSA.n.SIZE}$ results in setting $STATUS.CFGERR$ and ignoring the command.

Watermark regions are active only when $CTRLA.ENABLE=1$. If $CTRLA.ENABLE=0$, watermark regions are treated as exempt.

Figure 13-4. WATERMARK REGION WITH ONLY A SECURE SUBREGION

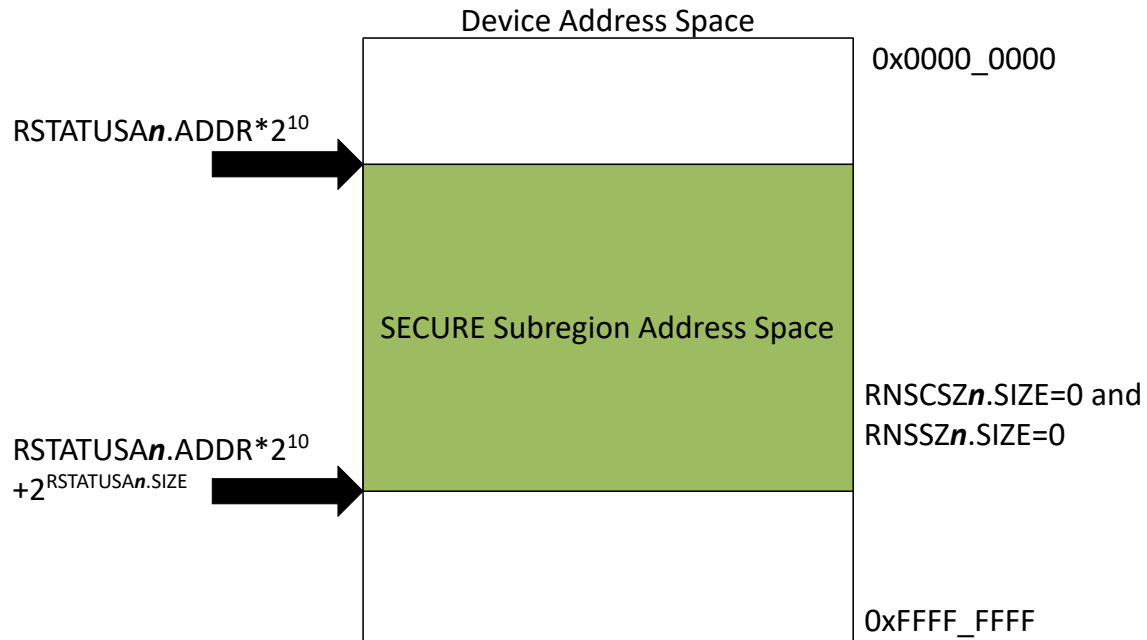


Figure 13-5. WATERMARK REGION WITH ONLY A NON-SECURE SUBREGION

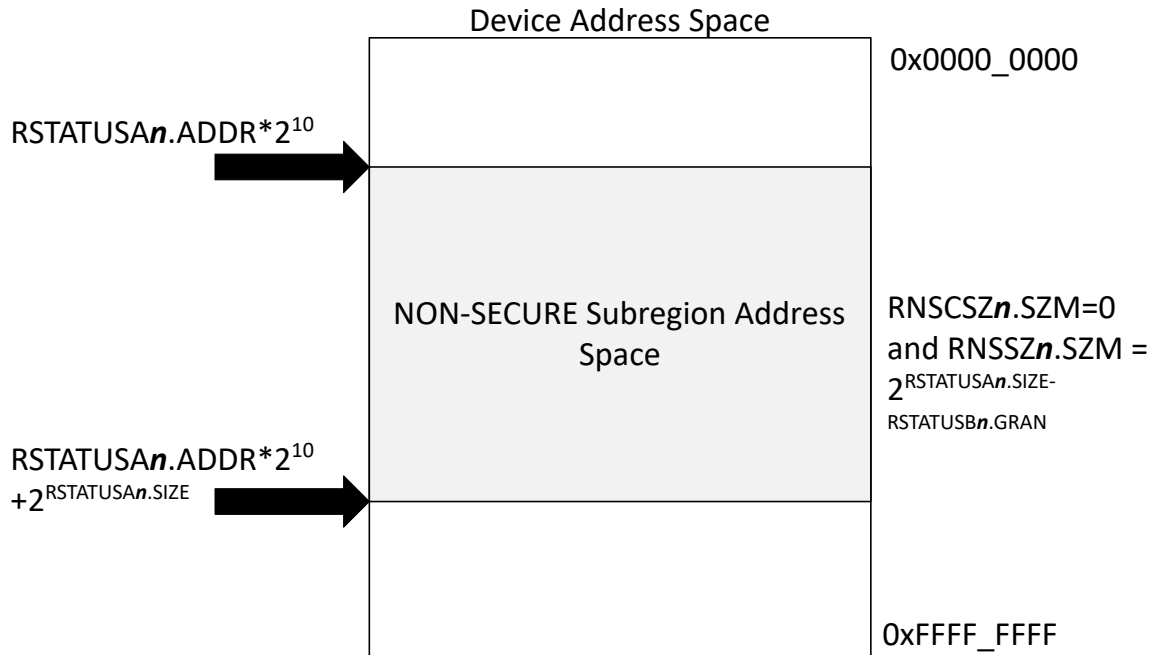


Figure 13-6. Watermark Region with a Secure and Non-Secure Callable Subregion

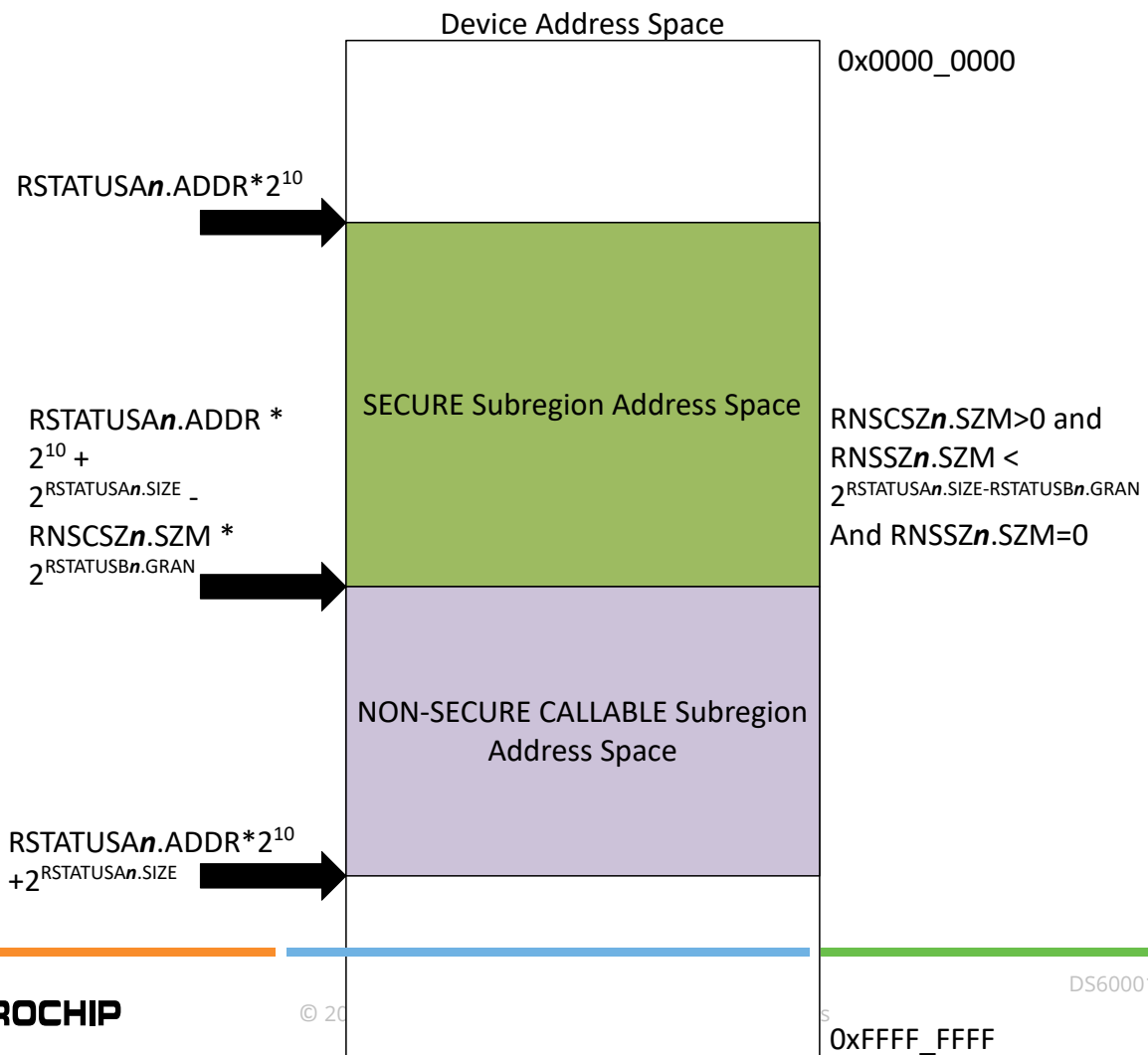
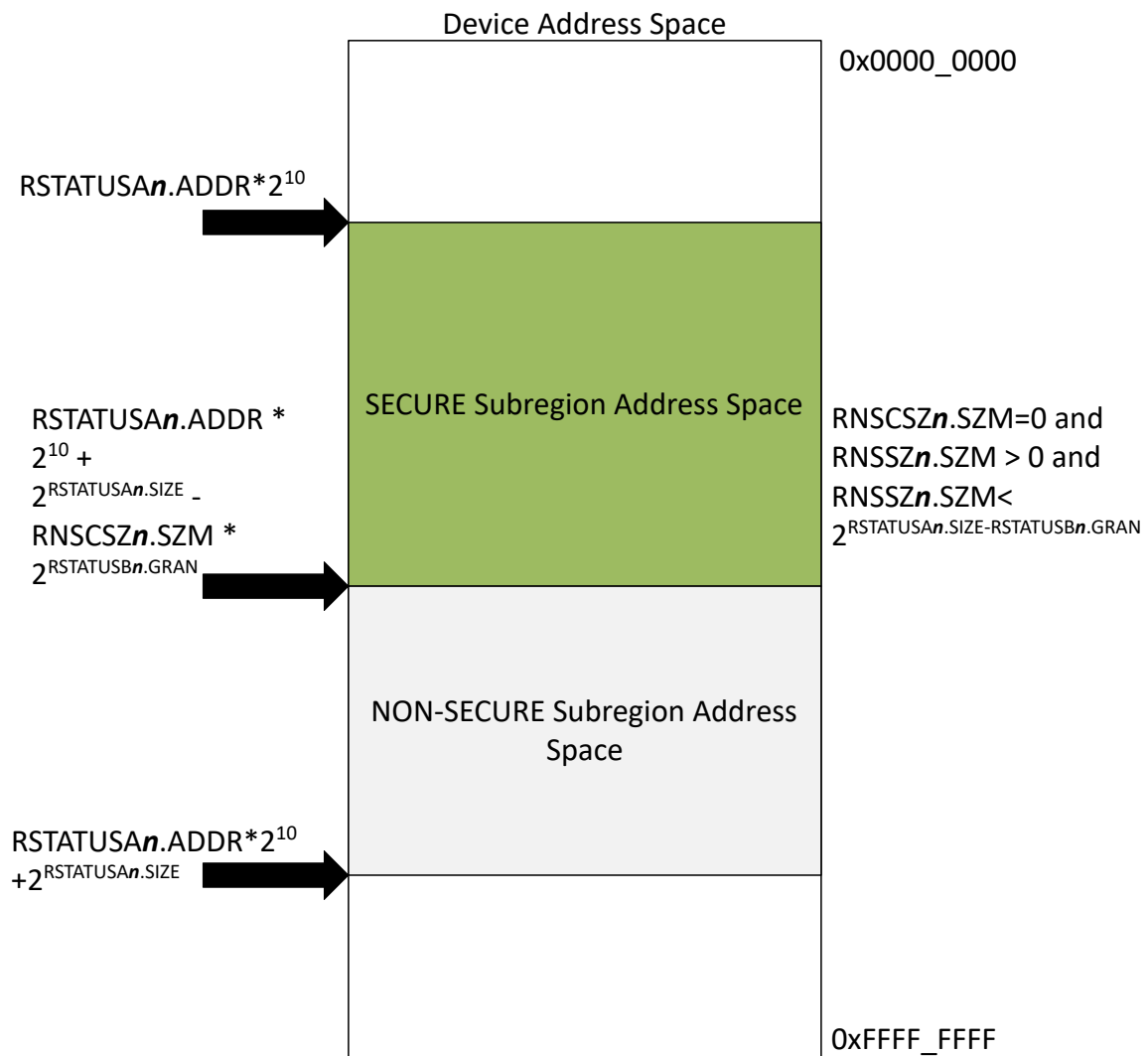


Figure 13-7. Watermark Region With a Secure and Non-Secure Subregion**Figure 13-8.** Watermark Region with a Non-Secure Callable and Non-Secure Subregion

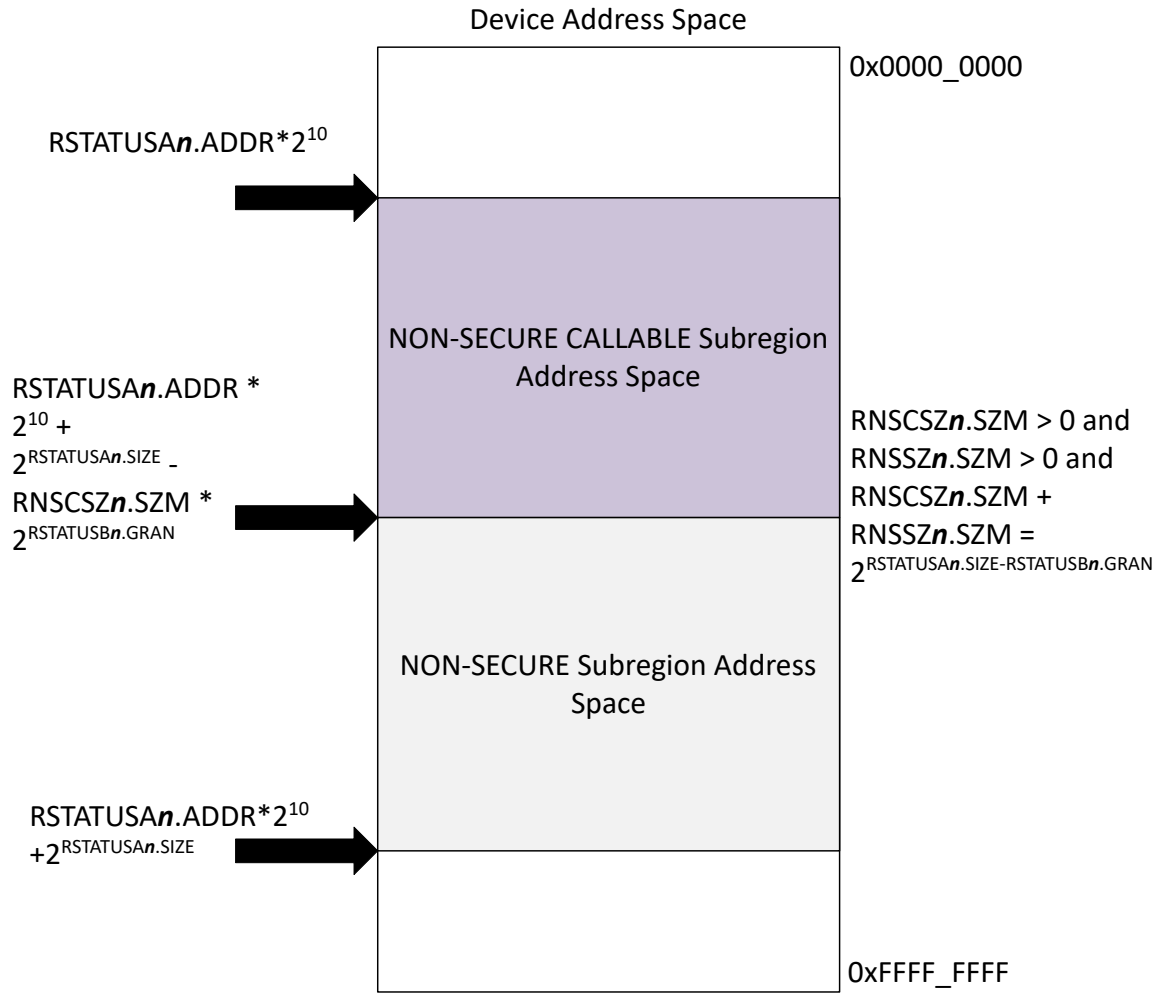
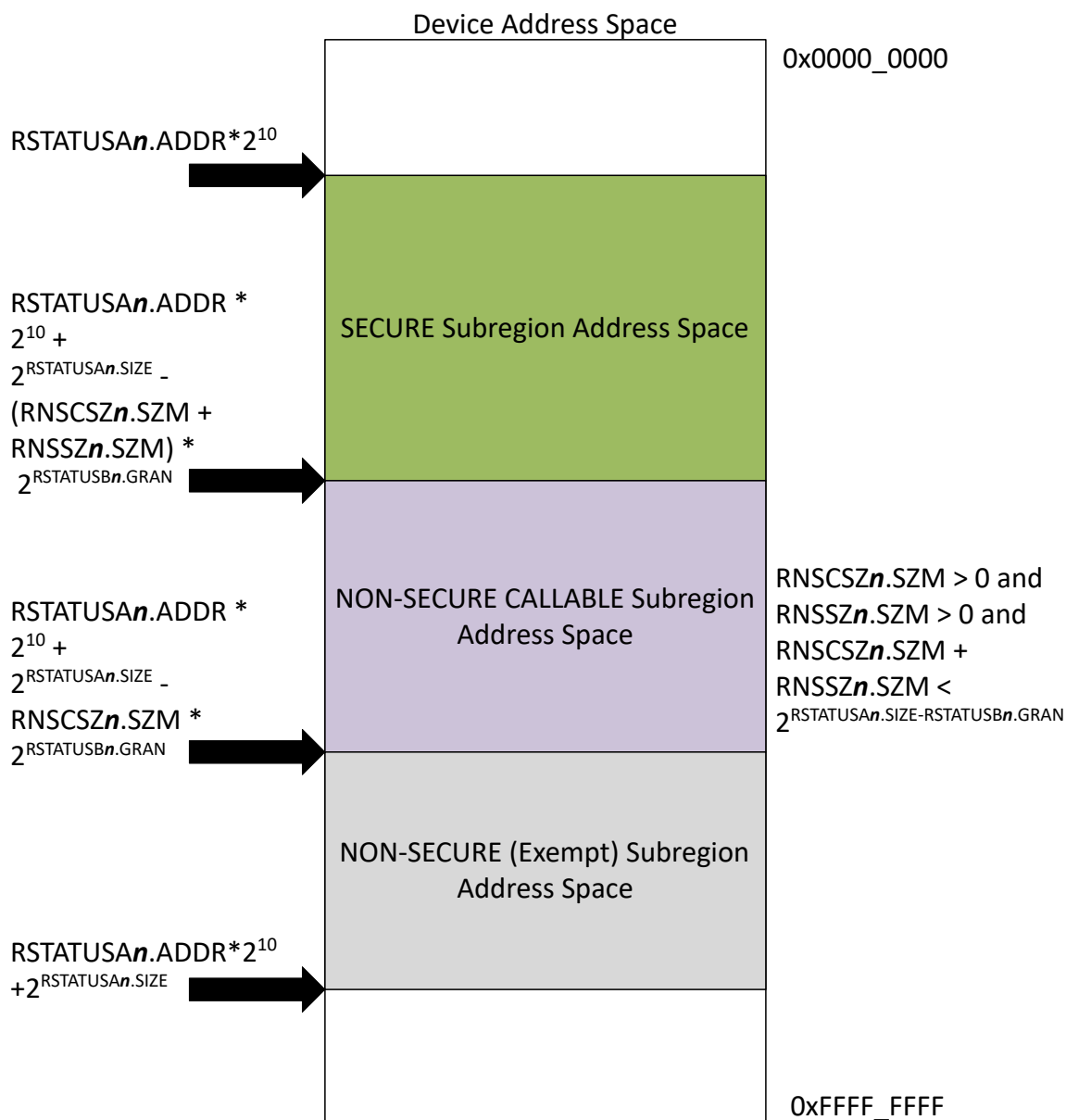


Figure 13-9. Watermark Region with all three Subregions



13.7 H2PB Exempt Main Regions

The Secure vs Non-Secure partitioning of the H2PB0/1/2 memory space is done in the H2PB bridge through the NONSEC registers. The partitioning is done peripheral by peripheral, bridges included. Each peripheral can be configured as Secure or Non-Secure, except PORT and EIC which can be configured as Mix-Secure or Non-Secure. This configuration only concerns PIC32CMMSG devices and is done through the BOOTCFG fuses and applied by the Rom code.

13.8 IDAU Regions Configuration

The device Boot ROM is in charge of configuring the module. All IDAU regions settings MUST be initialized prior to enabling the module. For PIC32CMGC devices, the BOOTCFG IDAU fuses are ignored and the configuration is hardcoded by the system and locked. For PIC32CMMSG devices, the ROM code will configure the IDAU based on the BOOTCFG IDAU fuses user configuration, which will allow to configure the sizes of the different IDAU regions for BFM, PFM and SRAM. Through the

fuses, the user also has the option to write lock the IDAU register, so that the configuration is locked till the next fuse update and device reboot.

Each IDAU region has its own register group composed of RCTRL, RSTATUSA, and RSTATUSB registers. The number of implemented register groups is readable from STATUSA.NBRG and cannot exceed 256.

To configure an IDAU region, first read its RSTATUSA register to determine its type. The type will determine which commands are legal and legal argument values.

Table 13-3. RSTATUS[A,B]

IDAU Region Type: RSTATUSA.TYPE	RSTATUSA.GRAN	RSTATUSA.MAXSZ	RSTATUSB.[SIZE, NONSEC]
INVALID	NA(reads 0)	NA(reads 0)	NA(reads 0)
SEC	NA(reads 0)	NA(reads 0)	NA(reads 0)
NONSEC_W	CFG Size in Bytes of one RCTRL[w].ARGLSb	CFG MAXSZ-1 is the maximum value of RCTRL[w].ARG	IDAU region size in Bytes
NSC_W	CFG Size in Bytes of one RCTRL[w].ARGLSb	CFG MAXSZ-1 is the maximum value of RCTRL[w].ARG	IDAU region size in Bytes
EXEMPT_B	CFG Size in Bytes of a block	CFG Maximum number of blocks inside the region	N/A. Secure versus Non-Secure configuration is applied in H2PB NONSEC register.

Notes:

1. **Bolded** cells denote static configuration whereas standard cells denote a configuration that can be modified using RCTRL commands.
2. The _W suffix indicates the IDAU region has a watermark configuration.
3. **CFG**: Hardcoded value to be read in the RSTATUSA register.

Table 13-4. Command and Arguments

RSTATUSA.TYPE	Valid RCTRL.CMD commands	RCTR.ARG Valid when
INVALID	None	NA
NONSEC_W	WRSZ	< RSTATUSA.MAXSZ
NSC_W	WRSZ	< RSTATUSA.MAXSZ ⁽¹⁾
EXEMPT_B	Reserved ⁽²⁾	Reserved ⁽²⁾

For each IDAU region, the command and argument validity are checked by the macro. When a command or argument is not valid:

- The command is discarded
- An error is reported to STATUSB.CFGERR

Before writing CTRL.ENABLE = 1:

- All configurable settings must be written (see the non-bolded cells in the RSTATUS[A,B] table)
- STATUSB.CFGERR must be read 0

The CTRL.CMD ENABLE command is discarded when STATUSB.CFGERR is high.

Once STATUSA.ENABLE is set, configuration is applied at the system level.

Notes:

1. The NSC_W size can exceed the remaining space size left by the NONSEC IDAU region in the MAIN region. In this case, no CFGERR error is reported and the NSC_W region overlaps exactly the whole SEC IDAU region, the NSC_W region cannot overlap MAIN regions it does not belong to.
2. The EXEMPT_B IDAU region type corresponds to the H2PB0/1/2 Main Regions. These regions are not configurable through the IDAU. For PIC32CMMSG devices, the peripherals (bridges included) inside these regions can be defined as Secure/Non-Secure (or Mix-secure/Non-Secure for EIC and PORT) through the BOOTCFG H2PB fuses. The configuration is applied by the ROM code.

13.9 Configuration Write Lock

The CTRL and all RCTR registers are write-protected after writing CTRL.CMD = WLCK. The write protection status is reflected by STATUSA.WLCK. Only a module reset can revert the write-protection.

These registers are also write-protected by the PAC.

STATUSB has no write-protection: it can be cleared on write when the configuration is locked and/or the module is PAC write-protected.

13.10 Peripheral Dependencies

Table 13-5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
IDAU	0x44012000		APB: CLKMSK[1] MASK2	---	24 STATUS0 PERID24			PD_CORE_SW

13.11 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	31:24	CMD[15:8]							
		23:16	CMD[7:0]							
		15:8								
		7:0								
0x04	STATUSA	31:24								
		23:16								NBRG[8]
		15:8	NBRG[7:0]							
		7:0							WLCK	ENABLE
0x08	STATUSB	31:24								
		23:16								
		15:8								
		7:0								CFGERR
0x0C ... 0x0FFF	Reserved									
0x1000	RCTRL	31:24								
		23:16								
		15:8								
		7:0								
0x1004	RSTATUSA	31:24								
		23:16				MAXSZ[5:0]				
		15:8				GRAN[5:0]				
		7:0				TYPE[3:0]				
0x1008	RSTATUSB	31:24								
		23:16								
		15:8								
		7:0								
0x100C	RSTATUSC	31:24								
		23:16								
		15:8								
		7:0								

13.11.1 Control

Name: CTRL
Offset: 0x0000
Reset: 0x00000000
Property: W

Note: This register does not apply for H2PB0/1/2 Main regions, that is IDAU EXEMPT_B regions with indices 10, 11, 12. The Secure/Non-Secure configuration for the peripherals (H2PB bridges included) is done through the BOOTCFG H2PB fuses and applied by the Rom code.

Table 13-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CMD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CMD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:16 – CMD[15:0] Command Register

The commands below apply to IDAU region types suffixed _W. Illegal commands are reported to STATUSB.CFGERR.

Value	Name	Description
0xA501	ENABLE	Module Enable
0xA502	DISABLE	Module Disable
0xA503	WLCK	Write Lock

13.11.2 Status A

Name: STATUSA
Offset: 0x0004
Reset: None
Property: R

Table 13-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								NBRG[8]
Access								R
Reset								
Bit	15	14	13	12	11	10	9	8
								NBRG[7:0]
Access	R	R	R	R	R	R	R	R
Reset								
Bit	7	6	5	4	3	2	1	0
							WLCK	ENABLE
Access							R	R
Reset								

Bits 16:8 – NBRG[8:0] Number Of IDAU Regions

Indicates the number of IDAU Regions.

The Region register group composed of RCTRL, RSTATUS[A,B] is repeated NBRG+1 times.

Region0 is always present and always of MR_INVALID type.

Regions1 to NBRG configuration and status registers are accessible through the IDAU Region register groups.

Bit 1 – WLCK Write Lock

Value	Description
0	The module is not Write-protected.
1	The module is Write-protected until next reset, writes are discarded and reported as PAC errors.

Bit 0 – ENABLE Enable

Value	Description
0	The module is disabled, all regions are considered exempt from security except those configured as valid when disabled (refer to the IDAU_MAIN_REGIONS_MAP parameter).
1	The module is enabled, dynamic configuration is applied to the system.

13.11.3 Status B

Name: STATUSB
Offset: 0x0008
Reset: 0x00000000
Property: R/K

Note: This Register is not write-protected.

Table 13-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								CFGERR
Access								
Reset								0

Bit 0 – CFGERR Configuration Error

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the CFGERR bit.

This bit is set when an invalid command is issued (from CTRL or RCTRL[x] registers). This bit is set when a valid command is issued with an invalid argument.

13.11.4 Region Control

Name: RCTRL
Offset: 0x1000
Reset: 0x00000000
Property: W

Note: This register does not apply for H2PB0/1/2 Main regions, that is IDAU EXEMPT_B regions with indices 10, 11, 12. The Secure/Non-Secure configuration for the peripherals (H2PB bridges included) is done through the BOOTCFG H2PB fuses and applied by the Rom code.

Table 13-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

13.11.5 Region Status A

Name: RSTATUSA
Offset: 0x1004
Reset: 0x00000000
Property: R

Table 13-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			MAXSZ[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
			GRAN[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					TYPE[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bits 21:16 – MAXSZ[5:0] Command Argument Maximum Size

Bits 13:8 – GRAN[5:0] Region Granularity

_W suffixed types (Watermark based regions): Size in Bytes of one RCTRL[x].ARG LSb.

_B suffixed types (Block based regions): Size in Bytes of a block.

Value	Name	Description
0x0	GRAN_NA	Not Applicable
0x5	GRAN_32B	
0x6	GRAN_64B	
0x7	GRAN_128B	
0x8	GRAN_256B	
0x9	GRAN_512B	
0xA	GRAN_1KB	
0xB	GRAN_2KB	
0xC	GRAN_4KB	
0xD	GRAN_8KB	
0xE	GRAN_16KB	
0xF	GRAN_32KB	
0x10	GRAN_64KB	
0x11	GRAN_128KB	

Value	Name	Description
0x12	GRAN_256KB	
0x13	GRAN_512KB	
0x14	GRAN_1MB	
0x15	GRAN_2MB	
0x16	GRAN_4MB	
0x17	GRAN_8MB	
0x18	GRAN_16MB	
0x19	GRAN_32MB	
0x1A	GRAN_64MB	
0x1B	GRAN_128MB	
0x1C	GRAN_256MB	
0x1D	GRAN_512MB	
0x1E	GRAN_1GB	
0x1F	GRAN_2GB	
0x20	GRAN_4GB	

Bits 3:0 – TYPE[3:0] IDAU Region Type

Value	Name	Description
0x0	INVALID	Region is invalid
0x1	SEC	Region is secure
0x2	NONSEC	Region is non secure
0x3	NONSEC_W	Region is non secure, the associated register group implements a watermark configuration
0x4	SECnNONSEC_B	Region is either secure or non-secure, the associated register group implements a block configuration
0x5	SECnNONSEC_L	Region is either secure or non-secure, configuration is linked to another IDAU region configuration bit (the linked configuration bit identification info is readable from the RSTATUSC register inside the same register group)
0x6	NSC_W	Region is non-secure callable, the associated register group implements a watermark configuration
0x7	EXEMPT_B	Region is exempt from security, the associated register group implements a block configuration

13.11.6 Region Status B

Name: RSTATUSB
Offset: 0x1008
Reset: 0x00000000
Property: R

This register contains the dynamic configuration of region 'n'.

This register is meaningful only for IDAU region types suffixed _B and _W, for other types it always reads 0.

Note: This register does not apply for H2PB0/1/2 Main regions, that is IDAU EXEMPT_B regions with indices 10, 11, 12.

Table 13-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

13.11.7 Region Status C

Name: RSTATUSC
Offset: 0x100C
Reset: 0x00000000
Property: R

Table 13-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

14. System Bus AHB - APB Bridge (H2PB)

14.1 Overview

The H2PB is a bridge between the AHB and APB buses. It allows for configuring or checking the Security setting for each APB Client Peripheral (H2PB itself included) and reports all access errors in-band.

14.2 Features

The following are key features of the H2PB module:

- All writes are treated as non-posted writes
- All errors are reported in-band to the initiator
- 1 NONSEC bit for each peripheral attached to each H2PB

14.3 Functional Description

The H2PB shares the AHB clock.

Each peripheral can be defined as Secure or Non-Secure through its related non-security NONSEC bit (refer to the peripheral dependency table for finding the index of each peripheral). Each NONSEC bit is included in 2 registers, one for setting it (NONSECSETA) and the other one for clearing it (NONSECCLRRA). Both registers will return the same value when being read.

When set, the CTRLA.SECEN bit enables the Trust Zone access control based on the NONSEC bits. Any non-secure access to a peripheral whose NONSEC bit is cleared will return an error. When CTRLA.SECEN is cleared, all accesses, secure or not, are allowed to pass through the bridge.

Device is configured with SECEN = 1 if Trust Zone is present and with SECEN = 0, if Trust Zone is not present.

For devices without Trust Zone, the NONSEC bits and H2PB registers are configured through the PAC. All Peripherals are then accessible (with no Secure vs Non-Secure notion).

The CTRLA.PRIV bit defines if non-privileged accesses are allowed to access the H2PB registers.

When access errors are detected (address holes, security violations, accesses to write protected registers), the errors are reported in-band to the initiator by the H2PB, or by the client peripheral through the H2PB.

14.3.1 Interrupts

The H2PB does not generate any interrupt. Access errors are reported in-band.

14.3.2 Events

The H2PB does not use and does not generate events.

14.3.3 Sleep Mode Operation

The H2PB stays active in active, idle, standby or sleepwalking modes when transactions occur on the AHB/APB buses. It will be off in backup, off and hibernate modes.

14.4 Peripheral Dependencies

Table 14-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
H2PB0	0x44032000		AHB: CLKMSK[0] MASK25 APB: CLKMSK[0] MASK26	---	19 STATUS0 PERID19			PD_CORE_SW
H2PB1	0x44838000		AHB: CLKMSK[1] MASK25 APB: CLKMSK[1] MASK26	---	46 STATUS1 PERID14			PD_CORE_SW
H2PB2	0x4500E000		AHB: CLKMSK[1] MASK27 APB: CLKMSK[1] MASK28	---	47 STATUS1 PERID15			PD_CORE_SW

14.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0					SECEN	PRIV		
0x04	WPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPLCK	WPEN
0x08	NONSECCLRA	31:24	NONSEC31	NONSEC30	NONSEC29	NONSEC28	NONSEC27	NONSEC26	NONSEC25	NONSEC24
		23:16	NONSEC23	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
		15:8	NONSEC15	NONSEC14	NONSEC13	NONSEC12	NONSEC11	NONSEC10	NONSEC9	NONSEC8
		7:0	NONSEC7	NONSEC6	NONSEC5	NONSEC4	NONSEC3	NONSEC2	NONSEC1	NONSEC0
0x0C	NONSECSETA	31:24	NONSEC31	NONSEC30	NONSEC29	NONSEC28	NONSEC27	NONSEC26	NONSEC25	NONSEC24
		23:16	NONSEC23	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
		15:8	NONSEC15	NONSEC14	NONSEC13	NONSEC12	NONSEC11	NONSEC10	NONSEC9	NONSEC8
		7:0	NONSEC7	NONSEC6	NONSEC5	NONSEC4	NONSEC3	NONSEC2	NONSEC1	NONSEC0
0x10	NONSECCLRB	31:24	NONSEC63	NONSEC62	NONSEC61	NONSEC60	NONSEC59	NONSEC58	NONSEC57	NONSEC56
		23:16	NONSEC55	NONSEC54	NONSEC53	NONSEC52	NONSEC51	NONSEC50	NONSEC49	NONSEC48
		15:8	NONSEC47	NONSEC46	NONSEC45	NONSEC44	NONSEC43	NONSEC42	NONSEC41	NONSEC40
		7:0	NONSEC39	NONSEC38	NONSEC37	NONSEC36	NONSEC35	NONSEC34	NONSEC33	NONSEC32
0x14	NONSECSETB	31:24	NONSEC63	NONSEC62	NONSEC61	NONSEC60	NONSEC59	NONSEC58	NONSEC57	NONSEC56
		23:16	NONSEC55	NONSEC54	NONSEC53	NONSEC52	NONSEC51	NONSEC50	NONSEC49	NONSEC48
		15:8	NONSEC47	NONSEC46	NONSEC45	NONSEC44	NONSEC43	NONSEC42	NONSEC41	NONSEC40
		7:0	NONSEC39	NONSEC38	NONSEC37	NONSEC36	NONSEC35	NONSEC34	NONSEC33	NONSEC32

14.5.1 H2PB CONTROL Register

Name: CTRLA
Offset: 0x0
Reset: 0x00000000
Property: R/W

Table 14-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					SECEN	PRIV		
Access					R/W	R/W		
Reset					0	0		

Bit 3 – SECEN When set, this bit enables the Trust Zone access control based on the NONSEC bits in the bridge SFRs and any non-secure access to a peripheral which has its NONSEC bit cleared will be rejected and returned as error to the AHB bus matrix. When cleared, this bit disables the Trust Zone access control and all accesses are allowed to pass though the bridge (Note for design: when this bit is cleared, the exported h2pb_nonsec vector is all one).

When set, this bit enables the Trust Zone access-control based on the NONSEC bits and any non-secure access to a peripheral which has its NONSEC bit cleared will be rejected and returned as an in-band error to the initiator.

When cleared, this bit disables the Trust Zone access-control and all accesses are allowed to pass though the bridge.

Bit 2 – PRIV When set, this bit will allow only privileged accesses to the bridge own SFRs. Any non-privileged read or write access to the bridge own SFRs will be rejected if this bit is set and will be returned as error to the AHB bus matrix.

When set, this bit will allow only privileged accesses to the H2PB registers. Any non-privileged read or write access to the bridge's own registers will be rejected if this bit is set and will be returned as an in-band error to the initiator.

14.5.2 H2PB Write Protect Control Register

Name: WPCTRL
Offset: 0x4
Reset: 0x00000000
Property: R/W

Table 14-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPLCK	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write protect command key. The desired value of WPKEY[23:0] is given by the parameter H2PB_WPCTRL_KEY. Writing a value other than H2PB_WPCTRL_KEY to this bit-field cancels the write operation to this register and generates a slave bus error. This bit-field always returns 0 on read.

Bit 1 – WPLCK 1 = WPCTRL register is write-protected (i.e. read-only) for non-debugger accesses. Non-debugger writes to this register are canceled and generate a slave bus error. This bit can only be cleared by a system hardware reset. 0 = WPCTRL register is not write protected. Once set, this bit will make WPCTRL write-protected (i.e. read-only), all the way until the next system hardware reset.

Bit 0 – WPEN Once set, this bit will make write-protected (i.e. read-only) all the SFRs in this macro which are marked by the property "Write-Protected" (like CTRLA, for example). WPCTRL itself is NOT affected by this bit because otherwise WPLCK would be totally redundant.

14.5.3 Non-Security Clear Register A

Name: NONSECCLRA
Offset: 0x8
Reset: 0x00000000
Property: R/K

Table 14-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NONSEC31	NONSEC30	NONSEC29	NONSEC28	NONSEC27	NONSEC26	NONSEC25	NONSEC24
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NONSEC23	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NONSEC15	NONSEC14	NONSEC13	NONSEC12	NONSEC11	NONSEC10	NONSEC9	NONSEC8
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NONSEC7	NONSEC6	NONSEC5	NONSEC4	NONSEC3	NONSEC2	NONSEC1	NONSEC0
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NONSEC H2PB_NONSEC_INIT_VAL[k] is the initial value of this bit at Reset. Writing a 1 to this bit clears the non-security bit for APB Slave k, k=0...31. Writing a 0 does nothing. Reading this bit delivers the current value of non-security bit of APB Slave k, k=0...31. The write command is accepted only if bit H2PB.WPCTRL.WPEN is not set. Accessing this register is also subject to security checks depending on the value of bit H2PB.CTRLA.SECEN. A write to any bit when the register is write-protected results in an error message being returned.

14.5.4 Non-Security SET Register A

Name: NONSECSETA
Offset: 0xc
Reset: 0x00000000
Property: R/S

Table 14-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NONSEC31	NONSEC30	NONSEC29	NONSEC28	NONSEC27	NONSEC26	NONSEC25	NONSEC24
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NONSEC23	NONSEC22	NONSEC21	NONSEC20	NONSEC19	NONSEC18	NONSEC17	NONSEC16
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NONSEC15	NONSEC14	NONSEC13	NONSEC12	NONSEC11	NONSEC10	NONSEC9	NONSEC8
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NONSEC7	NONSEC6	NONSEC5	NONSEC4	NONSEC3	NONSEC2	NONSEC1	NONSEC0
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NONSEC H2PB_NONSEC_INIT_VAL[k] is the initial value of this bit at Reset. Writing a 1 to this bit sets the non-security bit for APB Slave k, k=0...31. Writing a 0 does nothing. Reading this bit delivers the current value of non-security bit of APB Slave k, k=0...31 The write command is accepted only if bit H2PB.WPCTRL.WPEN is not set. Accessing this register is also subject to security checks depending on the value of bit H2PB.CTRLA.SECEN A write to any bit when this register is write-protected results in an error message being returned.

14.5.5 Non-Security Clear Register B

Name: NONSECCLRB
Offset: 0x10
Reset: 0x00000000
Property: R/K

Table 14-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NONSEC63	NONSEC62	NONSEC61	NONSEC60	NONSEC59	NONSEC58	NONSEC57	NONSEC56
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NONSEC55	NONSEC54	NONSEC53	NONSEC52	NONSEC51	NONSEC50	NONSEC49	NONSEC48
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NONSEC47	NONSEC46	NONSEC45	NONSEC44	NONSEC43	NONSEC42	NONSEC41	NONSEC40
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NONSEC39	NONSEC38	NONSEC37	NONSEC36	NONSEC35	NONSEC34	NONSEC33	NONSEC32
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NONSEC H2PB_NONSEC_INIT_VAL[k] is the initial value of this bit at Reset. Writing a 1 to this bit clears the non-security bit for APB Slave k, k=63...32. Writing a 0 does nothing. Reading this bit delivers the current value of non-security bit of APB Slave k, k=63...32. The write command is accepted only if bit H2PB.WPCTRL.WPEN is not set. Accessing this register is also subject to security checks depending on the value of bit H2PB.CTRLA.SECEN. A write to any bit when this register is write-protected results in an error message being returned.

14.5.6 Non-Security Set Register B

Name: NONSECSETB
Offset: 0x14
Reset: 0x00000000
Property: R/S

Table 14-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NONSEC63	NONSEC62	NONSEC61	NONSEC60	NONSEC59	NONSEC58	NONSEC57	NONSEC56
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NONSEC55	NONSEC54	NONSEC53	NONSEC52	NONSEC51	NONSEC50	NONSEC49	NONSEC48
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NONSEC47	NONSEC46	NONSEC45	NONSEC44	NONSEC43	NONSEC42	NONSEC41	NONSEC40
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NONSEC39	NONSEC38	NONSEC37	NONSEC36	NONSEC35	NONSEC34	NONSEC33	NONSEC32
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NONSEC H2PB_NONSEC_INIT_VAL[k] is the initial value of this bit at Reset. Writing a 1 to this bit sets the non-security bit for APB Slave k, k=63...32. Writing a 0 does nothing. Reading this bit delivers the current value of non-security bit of APB Slave k, k=63...32. The write command is accepted only if bit H2PB.WPCTRL.WPEN is not set. Accessing this register is also subject to security checks depending on the value of bit H2PB.CTRLA.SECEN A write to any bit when this register is write-protected results in an error message being returned.

15. Multi-Channel RAM Controller (MCRAMC)

15.1 Overview

The MCRAMC features ECC with Single Error Correction and Double Error Detection (SECEDED) for the data, with ECC fault injection for test purposes and ECC error status report.

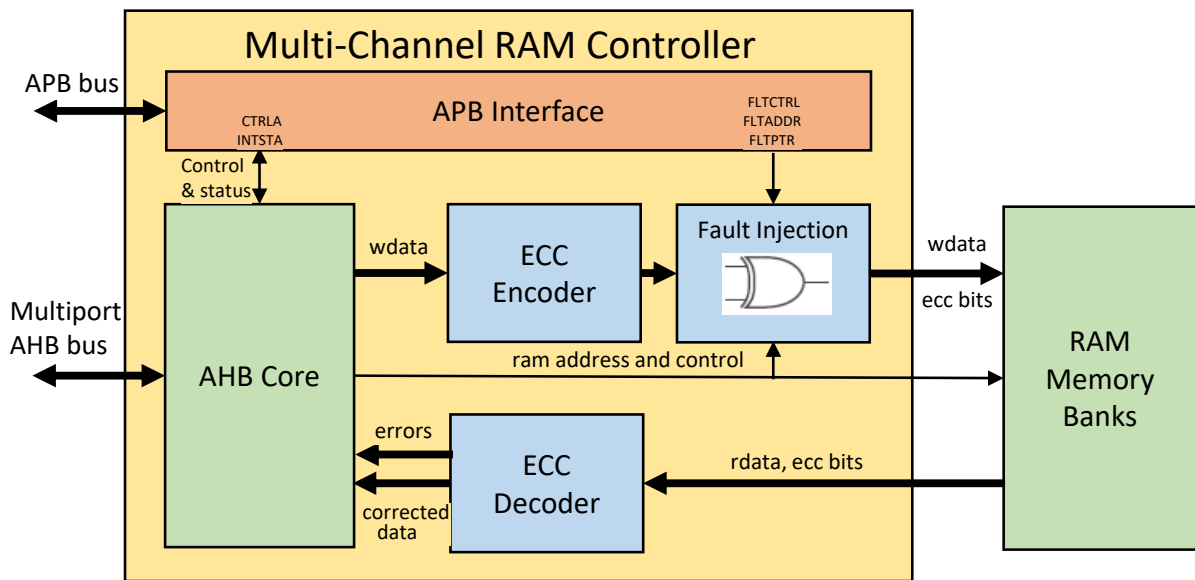
15.2 Features

Embedded Characteristics:

- ECC Single Error Correction on the fly
- ECC correction automatic write-back to RAM
- ECC synchronous abort exception for recoverable Double Error capability
- ECC test with fault injection
- ECC error status report and interrupt signaling

15.3 Block Diagram

Figure 15-1. MCRAMC Block Diagram



15.4 Functional Description

15.4.1 ECC SECEDED - ECC Functionality

The ECC decoding is enabled or disabled with the CTRLA.ENABLE bit. After reset the Rom code will copy the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse in the CTRLA.ENABLE bit.

The ECC feature can then be disabled and re-enabled using the same bit (for testing the ECC feature, for example, as detailed in the “ECC testing” chapter).

This disabling/enabling only applies to ECC decoding upon reads. The ECC encoding upon writes is always enabled and cannot be disabled.

Each 32-bit data word in SRAM is completed by 7 additional ECC bits, which are not accessible by the user.

Upon any 8, 16, or 32-bit write in the memory, the 7 ECC bits are computed and stored along with the data (The 8-bit or 16-bit writes are actually composed of an atomic read 32 bits, modify, write 32 bits).

Upon any 8, 16, or 32-bit read in the memory, if the ECC feature is disabled, then single or double errors are not detected, and not corrected. If the ECC feature is enabled, the ECC syndrome is computed on the related 32 data bits + 7 ECC bits.

Single-bit error read correction is performed on the fly with no penalty.

The MCRAMC writes back any corrected data into the RAM.

Simply reading the sensitive RAM content on a regular basis, for example from a DMA Host, prevents bit error accumulation. This is known as ECC memory scrubbing.

Double-bit error read detection triggers a bus error response from the MCRAMC, typically leading to a synchronous abort exception at the bus Host. This enables stopping of the bus Host access sequence precisely at the faulty address.

Upon a bus error response, the faulty address is registered in Error Capture Address Register.

If the corrupted RAM word laying at the 32-bit aligned address was originally provided from a non volatile memory, i.e., CPU instruction, it can be safely copied again into the RAM before returning from the abort exception handler. If the abort exception is immediately re-entered, there are 'stuck-at' bits in the corrupted memory word, which requires applying self-patching code technics prior returning from the abort exception handler.

Another example is if the corrupted RAM word was part of a DMA buffer from a communication peripheral, the communication might be retried if the DMA master has support for transfer error response.

15.4.2 SRAM Initialization

After a reset the SRAM content (data and ECC) is random and the ECC feature may be enabled depending on the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse value.

In case the ECC feature is enabled:

Any 32-bit write will initialize the data and the related ECC bits. However, 8-bit or 16-bit writes (which imply an internal read32/modify/write32) will probably trigger a single or double error on the internal 32-bit read (depending on the randomness of the 39 bits in memory). Consequently, the SRAM content must be initialized before it can be used.

The simplest option is to program a basic FOR loop filling the whole memory or to program a DMA transfer. The written data can take any value. However, care must be taken to only perform 32-bit writes in SRAM to access variables or DMA descriptors, and to not overwrite these data during the memory fill. The ECC bits will then be computed for each write and the memory will then be available for normal use.

15.4.3 ECC Testing

For ECC testing purpose, single-bit or double-bit faults can be injected during writes at a specific address. The address needs to be programmed in the MCRAMC Fault Injection Address Register (MCRAMC.FLTADDR), the bits to be flipped need to be programmed in the MCRAMC Fault Injection Pointer Register (MCRAMC.FLTPTR) and the fault injection needs to be enabled in the MCRAMC Fault Injection Control Register (MCRAMC.FLTCTRL).

Then the MCRAMC FLTCTRL Register has to be read prior to the first write access to the RAM.

After fault injections, if ECC decoding is disabled in the MCRAMC Control Enable A Register, reading at a faulty address directly shows the faulty data bits, if any. Then ECC decoding can be enabled again and fault injection disabled, to read and check single error correction or double error

detection. Then ECC decoding can be disabled again to read and check if a single error has been automatically corrected in the RAM array.

When ECC decoding is enabled in the MCRAMC Control Enable A Register, single-bit and double-bit ECC errors can be flagged in the MCRAMC INTSTA Interrupt Status register and can trigger an interrupt if enabled in the MCRAMC.INTENx Interrupt Enable register.

Note: This interrupt is an asynchronous CPU exception. It generally comes too late for system safe state recovery in case of a double-bit error.

The characteristics of a single-bit error are captured into the MCRAMC Error Capture registers and held stable until either the MCRAMC INTSTA.SER status bit is cleared by the software or a Double-bit error occurs.

The characteristics of a double-bit error are captured into the MCRAMC Error Capture registers and held stable until the MCRAMC INTSTA.DER status bit is cleared by the software.

The following constraints must be observed during the ECC testing process:

- After one or more writes to the MCRAMC user interface to change its configuration, a single read needs to be done from the MCRAMC FLTCTRL Register, prior to performing any access to the RAM.
- When both ECC decoding and fault injection are enabled, no single-bit fault RAM word must be read at the RAM fault injection address, because memory correction write-back would inject a fault again.
- When both ECC decoding and fault injection are enabled, a double-bit fault RAM word at the RAM fault injection address must be overwritten only with a 32-bit wide access.
- When fault injection is enabled, the data bits to be flipped, as programmed in the MCRAMC Fault Injection Pointer Register, must always be part of the bytes modified by the write access to the RAM fault injection address. A simple way to ensure this is to restrict the write accesses to the RAM fault injection address to be 32-bit wide only.

Note: A double error condition will generate a hard fault which will supersede the double error interrupt. If identification and soft recovery is desired, this condition must be detected in the hard fault handler, allowing the function to exit and the ISR for the double error to be executed.

15.4.4 Throughput Optimization

The throughput of the MCRAMC can be optimized when the system bus clock frequency is in lower range by removing any wait-state in the access to the RAM.

At higher frequency a wait-state must be inserted at the start of a burst.

When the MCRAMC_CTRLA.WS = 1, a wait state is inserted.

Refer to the *Electrical Characteristics* chapter for the configuration of WS.

15.4.5 Privileged Access

When the MCRAMC_CTRLA.PRIV = 1, the user interface access is granted only if the read or write access is performed only if the privileged attribute is set.

15.5 Peripheral Dependencies

Table 15-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
MCRAMC	0x4402C000	34 : INT	APB: CLKMSK[0] MASK21 AHB: CLKMSK[0] MASK22	---	16 STATUS0 PERID16			PD_CORE_SW

15.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	MCRAMC_CTRLA	31:24								
		23:16								
		15:8								WS
		7:0						PRIV	ENABLE	SWRST
0x04	Reserved									
...										
0x07										
0x08	MCRAMC_INTENCLR	31:24								
		23:16								
		15:8								
		7:0							DERREN	SERREN
0x0C	MCRAMC_INTENSET	31:24								
		23:16								
		15:8								
		7:0							DERREN	SERREN
0x10	MCRAMC_INTSTA	31:24								
		23:16								
		15:8								
		7:0							DERR	SERR
0x14	MCRAMC_FLTCTRL	31:24								
		23:16								
		15:8			FLTMD[1:0]					
		7:0							FLTEN	
0x18	MCRAMC_FLTPTR	31:24								
		23:16	FLT2PTR[7:0]							
		15:8								
		7:0	FLT1PTR[7:0]							
0x1C	MCRAMC_FLTADR	31:24								
		23:16	FLTADR[23:16]							
		15:8	FLTADR[15:8]							
		7:0	FLTADR[7:0]							
0x20	MCRAMC_ERRCADR	31:24								
		23:16	ERCADR[23:16]							
		15:8	ERCADR[15:8]							
		7:0	ERCADR[7:0]							
0x24	MCRAMC_ERRCPAR	31:24								
		23:16								
		15:8								
		7:0	ERCPAR[7:0]							
0x28	MCRAMC_ERRCSYN	31:24								
		23:16								
		15:8	ERR2	ERR1						
		7:0	ERCSYN[7:0]							
0x2C	Reserved									
...										
0xFB										
0xFC	MCRAMC_VERSION	31:24								
		23:16						MFN[2:0]		
		15:8					VERSION[11:8]			
		7:0	VERSION[7:0]							

15.6.1 Control Enable A Register

Name: MCRAMC_CTRLA
Offset: 0x00
Reset: 0x00000002
Property: RW

Table 15-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
								WS
Access								
Reset								0
Bit	7	6	5	4	3	2	1	0
						PRIV	ENABLE	SWRST
Access								
Reset						0	1	0

Bit 8 – WS Wait State Enable

Bit 2 – PRIV Privileged Access Enable

Bit 1 – ENABLE ECC Decoder Enable

Value	Name	Description
0		ECC decoding is disabled.
1		ECC decoding is enabled.

Bit 0 – SWRST Software Reset

Value	Name	Description
0		No effect.
1		Reset the MCRAMC. A software-triggered hardware reset of the MCRAMC user interface is performed.

15.6.2 Interrupt Enable Clear Register

Name: MCRAMC_INTENCLR
Offset: 0x08
Reset: 0x00000000
Property: R/W/K

Table 15-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DERREN	SERREN
Access								
Reset							0	0

Bit 1 – DERREN Double Bit Error Interrupt Enable, Clear

Bit 0 – SERREN Single Bit Error Interrupt Enable, Clear

15.6.3 Interrupt Enable Set Register

Name: MCRAMC_INTENSET
Offset: 0x0C
Reset: 0x00000000
Property: R/W/S

Table 15-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DERREN	SERREN
Access								
Reset							0	0

Bit 1 – DERREN Double Bit Error Interrupt Enable, Set

Bit 0 – SERREN Single Bit Error Interrupt Enable, Set

15.6.4 Interrupt Status Register

Name: MCRAMC_INTSTA
Offset: 0x10
Reset: 0x00000000
Property: R/W/K

Table 15-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DERR	SERR
Access								
Reset							0	0

Bit 1 – DERR Double Bit Error

Bit 0 – SERR Single Bit Error

15.6.5 Fault Injection Control Register

Name: MCRAMC_FLTCTRL
Offset: 0x14
Reset: 0x00000000
Property: RW

Table 15-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			FLTMD[1:0]					
Access								
Reset			0	0				
Bit	7	6	5	4	3	2	1	0
							FLTEN	
Access								
Reset							0	

Bits 13:12 – FLTMD[1:0] Fault Injection Mode

Bit 1 – FLTEN Fault Injection Enabled

Value	Name	Description
0		Disables fault injection.
1		Enables fault injection at FLTADR address offset as selected by FLTMD and FLTxPTR.

15.6.6 Fault Injection Pointer Register

Name: MCRAMC_FLTPTR
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 15-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	FLT2PTR[7:0]							
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access	FLT1PTR[7:0]							
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – FLT2PTR[7:0] Double Fault Injection Bit Pointer

Bits 7:0 – FLT1PTR[7:0] Single Fault Injection Bit Pointer

15.6.7 Fault Injection Address Register

Name: MCRAMC_FLTADR
Offset: 0x1C
Reset: 0x00000000
Property: RW

Table 15-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	FLTADR[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FLTADR[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FLTADR[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – FLTADR[23:0] Fault Address Offset

15.6.8 Error Capture Address Register

Name: MCRAMC_ERRCADR
Offset: 0x20
Reset: 0x00000000
Property: R

Table 15-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	ERCADR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ERCADR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ERCADR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – ERCADR[23:0] ECC SECDED Error Capture Address

15.6.9 Error Capture Parity Register

Name: MCRAMC_ERRCPAR
Offset: 0x24
Reset: 0x00000000
Property: R

Table 15-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	ERCPAR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ERCPAR[7:0] ECC SECDED Error Capture Parity

15.6.10 Error Capture Syndrome Register

Name: MCRAMC_ERRCSYN
Offset: 0x28
Reset: 0x00000000
Property: R

Table 15-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ERR2	ERR1						
Access	R	R						
Reset	0	0						
Bit	7	6	5	4	3	2	1	0
	ERCSYN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – ERR2 ECC Double Bit Error

Value	Name	Description
0		Not a Double bit error.
1		Double bit error.

Bit 14 – ERR1 ECC Single Bit Error

Bits 7:0 – ERCSYN[7:0] ECC SECDED Error Capture Syndrome

Value	Name	Description
0		Not a Single bit error.
1		Single bit error.

15.6.11 Version Register

Name: MCRAMC_VERSION
Offset: 0xFC
Reset: 0x00000510
Property: R

Table 15-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
						MFN[2:0]		
Access						R	R	R
Reset						0	0	0
Bit	15	14	13	12	11	10	9	8
					VERSION[11:8]			
Access					R	R	R	R
Reset					0	1	0	1
Bit	7	6	5	4	3	2	1	0
	VERSION[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 18:16 – MFN[2:0] Metal Fix Number

Bits 11:0 – VERSION[11:0] MCRAMC Version

16. Peripheral Access Controller (PAC)

16.1 Overview

The Peripheral Access Controller provides an interface for the locking and unlocking of predefined peripheral registers within the device. It controls reporting of all violations that could happen when accessing a peripheral: write protected access, illegal access, enable protected access, access when clock synchronization or software reset is on-going. If access errors are reported in-band then the system uses bus errors (aka bus faults). If errors are not reported in-band then the PAC reports them via a unique interrupt flag per peripheral. The PAC module also reports errors occurring at the client bus level, when an access to a non-existing address is detected.

16.2 Features

The following are key features of the PAC module:

- Programmable write protect bit for each peripheral that supports it
- Interrupt generation on access errors to peripherals
 - Write protect error
 - Unused Address
 - Access when clock synchronization is active
 - Access when software reset is on-going
 - Access to reserved memory regions

16.3 Debug Operation

When the CPU is halted in Debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

16.4 Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Write Control (WRCTRL) register
- AHB Slave Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

16.5 Functional Description

16.5.1 Basic Operation

16.5.1.1 Operations

The PAC module allows the user to set, clear, or lock the write protected status of any peripherals. If a peripheral access violation occurs, the Peripheral Interrupt Flag registers (INTFLAGx, x=A, B, C, or D) are updated to inform the user which peripherals have had access violations. The corresponding Peripheral Write Control Status n register (STATUSx) gives the state of the write protection for all peripherals.

16.5.1.2 Peripheral Access Errors

The following events will generate a Peripheral Access Error:

- **Protected write:** To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as "PAC Write-Protection" in the module's data sheet can be protected. If a peripheral is not write protected, write data accesses are performed normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in one of the INTFLAGx register will be set.
- **Illegal access:** Access to an unimplemented register within the module.
- **Synchronized write error:** For write-synchronized registers an error will be reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGx registers bits are set, an interrupt will be requested if the PAC interrupt enable bit is set.

16.5.1.3 Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register. The data written to the WRCTRL register is composed of two fields: WRCTRL.PERID and WRCTRL.KEY. The WRCTRL.PERID is a unique identifier corresponding to a peripheral (The value corresponding to each peripheral can be found in the peripheral dependency table) . The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be 'clear protection', 'set protection', and 'set and lock protection bit'.

- The clear protection operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.
- The set protection operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.
- The set and lock protection operation will set the write access protection for the peripheral selected by WRCTRL.PERID and locks the access rights of the selected peripheral registers. The write access protection will only be cleared by a hardware reset.

Any peripheral access control status can be read from the corresponding STATUSn register and its lock state from LOCKn register.

16.5.1.4 Write Access Protection Management Errors

Only word-wise (or register-wise) writes to the WRCTRL register will change the access protection. Other types of accesses will have no effect and will cause a PAC write access error.

The PAC also generates an in-band error in the following conditions:

- If a peripheral is write protected and a subsequent set protection or set and lock operation is detected then the PAC returns an error.
- If a peripheral is not protected and a subsequent clear protection operation is detected then the PAC returns an error.
- If a peripheral is write protected and locked and either a clear protection or set protection operation is detected, then the PAC returns an error.

This can be used to ensure that the application follows any write protect with an unprotect and any unprotect with a protect. In applications where the write protect state of a peripheral is manipulated in several contexts, care should be taken so that the contexts do not interfere with each other.

16.5.2 Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available bus host (CPU, DMA) is running. The PAC will continue to catch access errors from the module and generate interrupts or events.

16.6 Peripheral Dependencies

Table 16-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
PAC	0x44022000		APB: CLKMSK[0] MASK14	---	14 STATUS0 PERID14			PD_CORE_SW

16.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0						PRIV		
0x04	WRCTRL	31:24								
		23:16	KEY[7:0]							
		15:8								
		7:0	PERID[7:0]							
0x08	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								ERREO
0x0C ... 0x17	Reserved									
0x18	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								ERR
0x1C	INTENSET	31:24								
		23:16								
		15:8								
		7:0								ERR
0x20	INTFLAG0	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x24	INTFLAG1	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x28	INTFLAG2	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x2C	INTFLAG3	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x30	INTFLAGSET0	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x34	INTFLAGSET1	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x38	INTFLAGSET2	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x3C	INTFLAGSET3	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x40	STATUS0	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x44	STATUS1	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x48	STATUS2	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x4C	STATUS3	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x50	LOCK0	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x54	LOCK1	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x58	LOCK2	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
0x5C	LOCK3	31:24	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
		23:16	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
		15:8	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
		7:0	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0

16.7.1 Control A Register

Name: CTRLA
Offset: 0x0
Reset: 0x00000000
Property: R/W

Table 16-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						PRIV		
Access						R/W		
Reset						0		

Bit 2 – PRIV Privileged Access Only

Value	Name	Description
1	ENABLE	Macro registers only accessible in privileged accesses
0	DISABLE	Macro register accessible in privileged and unprivileged accesses.

16.7.2 Write Control Register

Name: WRCTRL
Offset: 0x4
Reset: 0x00000000
Property: W

Table 16-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	KEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PERID[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – KEY[7:0] These bits define the peripheral access control key: others = reserved Note: Always reads 0's. Note: Bits 7-2 are ignored and maybe unimplemented
These bits define the peripheral access control key:

Value	Name	Description
3	LOCK	LOCK - Set and Lock the write protection state of the peripheral until the next reset
2	SET	SET - Set the peripheral write protection
1	CLEAR	CLEAR - Clear the peripheral write protection
0	OFF	OFF - No Action

Bits 7:0 – PERID[7:0] The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. The Peripheral Identifier is provided by the chip spec and corresponds to the PERID[7:0]. For INTFLAGn.PERIDm, STATUSn.PERIDm and LOCKn.PERIDm, n=PERID/32 and m=PERID%32. Note: Always reads 0's. Note: Bit 7 is ignored and maybe unimplemented.

The PERID represents the peripheral whose control is changed using the WRCTRL.KEY.
The value corresponding to each peripheral can be found in the peripheral dependency table.

16.7.3 Event Control Register

Name: EVCTRL
Offset: 0x8
Reset: 0x00000000
Property: R/W

Table 16-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								ERREO
Access								R/W
Reset								0

Bit 0 – ERREO This bit indicates if the Peripheral Access Error Event Output is enabled or disabled. When enabled, the event is asserted while any INTFLAGn bit is set:

Value	Name	Description
1		Peripheral Access Error Event Output is enabled.
0		Peripheral Access Error Event Output is disabled.

16.7.4 Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x18
Reset: 0x00000000
Property: R/K

Table 16-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/K
Reset								0

Bit 0 – ERR Writing a '1' to this field clears the interrupt enable.

Value	Name	Description
1		Interrupt enabled
0		Interrupt disabled

16.7.5 Interrupt Enable Set Register

Name: INTENSET
Offset: 0x1c
Reset: 0x00000000
Property: R/S

Table 16-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								ERR
Access								R/S
Reset								0

Bit 0 – ERR Writing a '1' to this field sets the interrupt enable.

Value	Name	Description
1		Interrupt enabled
0		Interrupt disabled

16.7.6 Interrupt Flag n Register

Name: INTFLAG
Offset: 0x20 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: R/K

This flag is set when an access error is detected by the CLIENT n, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the corresponding INTFLAGAHB interrupt flag.

Notes:

1. Not available on 100-pin packages
2. Not available on 100-pin and 144-pin packages
3. Not available on 100-pin, 144-pin, and 176-pin packages

Table 16-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
Access	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
Access	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
Access	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
Access	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PERID This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAG bit, and generates an interrupt request if INTENCLR/SET.ERR is one. Read value reflects the state of the interrupt flag.

Value	Name	Description
1		Interrupt pending
0		Interrupt not pending

16.7.7 Interrupt Flag Set n Register

Name: INTFLAGSET
Offset: 0x30 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: R/S

Table 16-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
Access	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
Access	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
Access	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
Access	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
– PERID This bit provides a software method to set the associate INTGLAGn bit so the ISR can be tested. This feature may be useful for safety applications. Read value reflects the state of the INTFLAGn. Write to '1' to set the corresponding bit in INTFLAGn.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

16.7.8 Peripheral Status n Register

Name: STATUS
Offset: 0x40 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: R

Reading the STATUS1 register returns the peripheral write protection status of the indicated peripherals:

0: Peripheral is not write protected.

1: Peripheral is write protected.

Note: The PERIDxx value corresponding to each peripheral can be found in the peripheral dependency table.

Table 16-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PERID PERID Write Protection Status

Value	Name	Description
1		Write Protection enabled
0		Write Protection disabled

16.7.9 Peripheral Lock n Register

Name: LOCK
Offset: 0x50 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: R

Table 16-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERID31	PERID30	PERID29	PERID28	PERID27	PERID26	PERID25	PERID24
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PERID23	PERID22	PERID21	PERID20	PERID19	PERID18	PERID17	PERID16
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PERID15	PERID14	PERID13	PERID12	PERID11	PERID10	PERID9	PERID8
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PERID7	PERID6	PERID5	PERID4	PERID3	PERID2	PERID1	PERID0
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PERID PERID Write Protection Lock Status

Value	Name	Description
1		Write Protection Locked
0		Write Protection Unlocked

17. Device Service Unit (DSU)

17.1 Overview

The Device Service Unit (DSU) provides a means to detect debugger probes and restrict their access. It also implements a Microchip Device ID and a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Therefore, it complies with the Arm Peripheral Identification specification.

The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix.

The DSU implements communication channels between the device and external tools which can be used at boot time to make use of Boot ROM services.

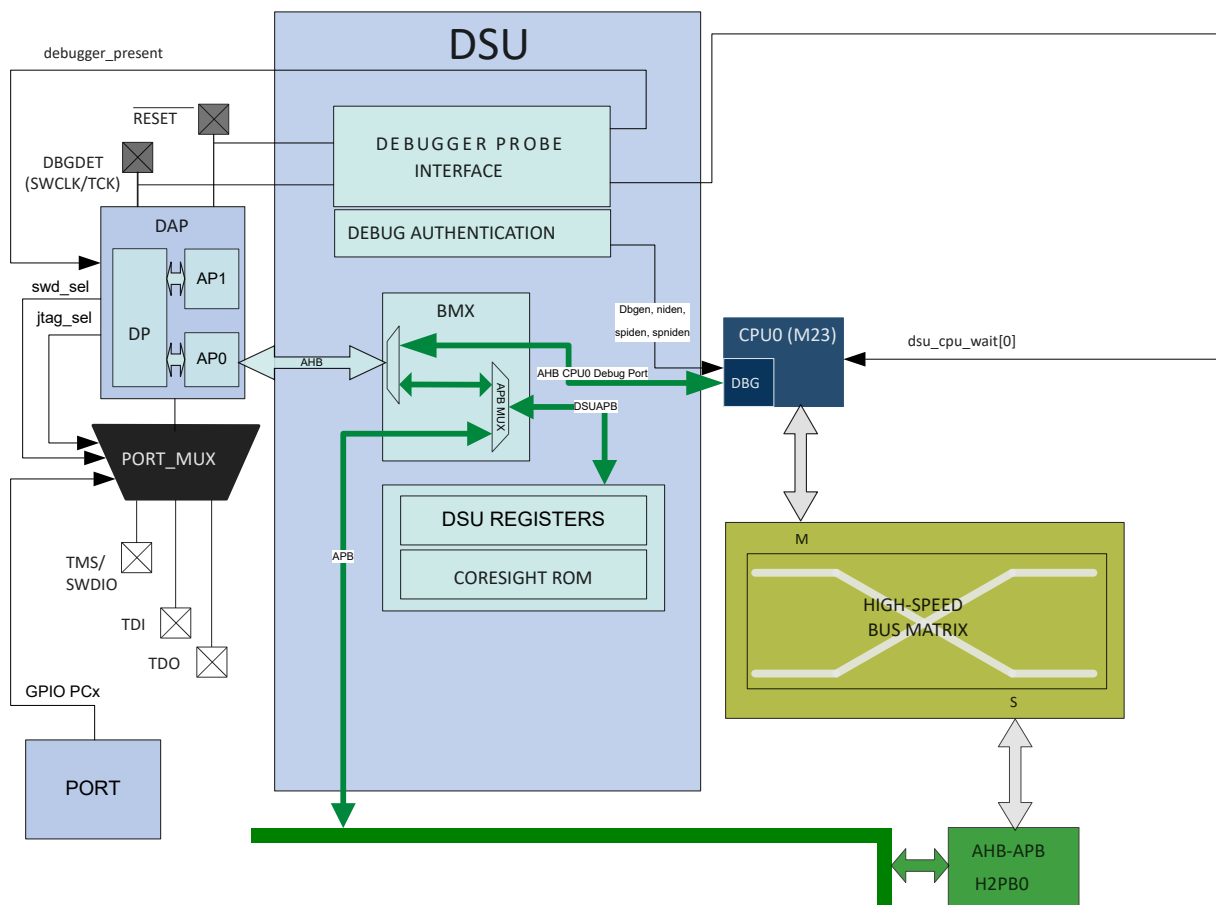
17.2 Features

The following are key features of the DSU module:

- Arm® CoreSight™ compliant device identification
- Microchip device identification
- CPU and Boot ROM reset extension
- Debugger probe detection (Cold-Plugging and Hot-Plugging)
- Two Debug Communication Channels (DCC[0,1] with DMA signaling)
- Two Boot Communication Channels (BCC[0,1])
- Supports Debug Authentication and non-secure debug

17.3 DSU Block Diagram

Figure 17-1. Block Diagram



17.4 Signal Description

The DSU uses these signals to function.

Signal Name	Type	Description
RESET	Digital Input	External Reset
SWCLK/TCK	Digital Input	Serial Wire Debug (SWD) clock / JTAG Test Clock
SWDIO/TMS	Digital I/O	Serial Wire Debug (SWD) bidirectional data pin / JTAG Test Mode Select
TDI	Digital Input	JTAG Test Data In
TDO	Digital Output	JTAG Test Data Out

Note:

- By default the Debug Port (DP) starts in JTAG mode after Power-on Reset, but it can switch to Single Wire Debug (SWD) mode using the Arm JTAG to SWD switching sequence.

17.5 Debug Operation

17.5.1 Principle of Operation

The DSU monitors the presence of a debugger to control appropriately on-chip resources when a debugger is attached to the device. The following are two attachment methods with different properties discussed later:

- Cold-plugging: Attachment when external reset is asserted.
- Hot-plugging: Attachment when external reset is not asserted.

A debugger presence has several effects on the device:

- Assigns JTAG or SWD pads to the Debug Port (DP) by controlling the port mux
- Enables Access Ports
- Extends the device CPU's Boot ROM and reset phases

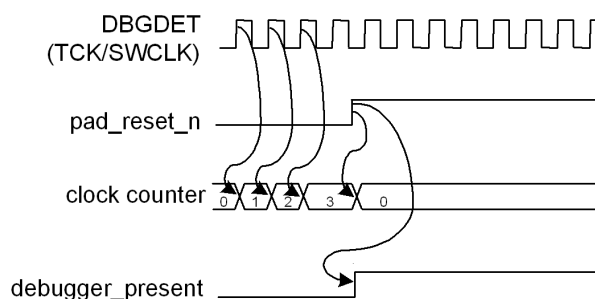
The DSU also implements Boot Communication Channels to enable communication with the device's Boot ROM. For additional information on the Arm debug components, refer to the Arm "Debug Interface v5 Architecture Specification" document.

17.5.2 Debugger Probe Detection

17.5.2.1 Cold Plugging

Cold-Plugging is the detection of a debugger when the external reset is asserted ($\overline{\text{RESET}}$ pulled low). As shown in the following figure, a cold-plugging is detected when at least 3 TCK/SWCLK pin rising edges are detected while $\overline{\text{RESET}}$ is asserted. The detector state is updated upon a $\overline{\text{RESET}}$ pin rising edge. The Cold-Plugging detection is reset by a Power-on Reset, Brown-out Reset, or external reset. At startup, the TCK/SWCLK pin is internally pulled up to avoid false detection of a debugger when this pin is left unconnected on the application board. Cold-plugging detection is available once pads are correctly powered after a power-reset. If DAL.CPU0 equals 0 or 3, then Cold-Plugging is the only way to detect a debugger probe, hence the external reset timing must be longer than the power-reset timing. If external reset is de-asserted before power-reset release, the user must retry the cold-plugging procedure until it gets connected to the device.

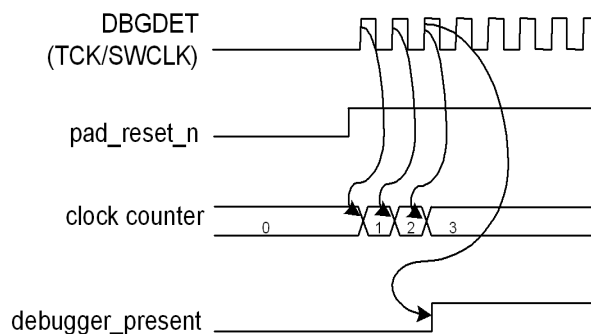
Figure 17-2. Cold-Plugging Timing Diagram



17.5.2.2 Hot Plugging

Hot-Plugging is the detection of a debugger probe when the external reset pin is not asserted. As shown in the following figure, Hot-Plugging is detected when at least 3 TCK/SWCLK pin rising edges are detected while $\overline{\text{RESET}}$ is not asserted. The hot-plugging detector is reset by a power-reset, a Brown-out Reset, or an external reset. For security reasons, Hot-Plugging is not available when DAL.CPU0 equals to 0 or 3. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Status bit of the Status B register (STATUSB.HPS).

Figure 17-3. Hot-Plugging Detection Timing Diagram



17.5.2.3 Debugger Probe Detection Conditions and Effects

Once the debugger is attached, the DSU takes ownership of the JTAG port pins to prevent application code from reclaiming them. If the debugger switches to SWD mode using the ARM JTAG to SWD switching sequence, the unused JTAG IOs are released, claiming the SWCLK and SWDIO IOs only. Once in SWD mode, the debugger can switch back to JTAG mode using the Arm SWD to JTAG switching sequence at which point the hardware will claim all the JTAG IOs again.

The DP starts in JTAG mode after a power-reset.

Following a debugger detection:

- The Debugger Present bit of the Status B register (STATUSB.DBGPRES) reads one
- The CPU reset gets extended (only with cold-plugging)

17.5.3 CPU RESET EXTENSION

17.5.4 BOOT ROM PHASE EXTENSION

Enter a short description of your concept here (optional).

17.5.5 Debugger Security

The DSU provides Debug Access Level (DAL) mechanism which disable or limit the debugger access into the device. The DAL settings are programed through the ROMCFG.DAL fuse configuration bits.

The DSU DAL feature provides three levels of access. The following table describes the debugger access levels:

Table 17-1. Debugger Access Level Through DAP

DAL Setting	DSU Address Space		System Memory Regions ⁽¹⁾		Hot Plugging	Cold Plugging
	External	Internal	Secure	Non-Secure		
DAL 0	X	-	-	-	-	X
DAL 1 ⁽¹⁾	X	-	-	X	X	X
DAL 2	X	X	X	X	X	X

Note:

1. Only available on PIC32CM SG devices. Secure requests are converted to non-secure request in the CPU.

If the debugger attempts to access a device region which is not permitted by the current DAL setting, a bus error is returned to the CoreSight AP hardware which results in the assertion of the ARM DP sticky error bit.

In DAL0 mode, debugger breakpoints are disabled.

17.5.6 Boot Communication Channels

Boot Communication Channels allow communication between a debug adapter and the Host CPU executing the Host Boot ROM at startup. The Host Boot ROM implements system level commands.

17.6 Programming

For each bus system, programming the Flash or RAM memories is only possible when the debugger access level is sufficient to access the desired resource. If MEM-APx is selected by the DP and if DAL.CPU0 is equal to:

- 0x2: debugger can access all areas in the CPU bus system.
- 0x0: If the debugger targets the MEM-AP0 then it can only access the DSU external address space making it possible to communicate with the Boot ROM after reset. If debugger targets the MEM-AP1 then all transactions results in a DAP fault.

A typical programming procedure, when DAL.CPU0 = 0x2, is presented below:

1. At power-up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until all the internally regulated supplies have reached a safe operating state.
2. The Power Manager (PM) starts, clocks are switched on (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal resets are maintained due to the external reset.
3. The debugger generates at least 3 SWCLK clock cycles while $\overline{\text{RESET}}$ is asserted. $\overline{\text{RESET}}$ is then released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. CPU 0 executes its Boot ROM.
6. It is recommended to issue a chip-erase (supported by the Boot ROM) to ensure that the Flash is fully erased prior to programming.
7. If the operation issued above was accepted and has completed successfully then DAL.CPU0 equals 0x2 therefore programming is available through the MEM-AP 0.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or sending a command to the Boot ROM to jump to the NVM code. Make sure that the SWCLK clock is stopped while asserting $\overline{\text{RESET}}$ to prevent entering again the cold-plugging procedure.

17.7 Device Identification

Device identification relies on the Arm CoreSight component identification scheme, which allows the chip to be identified as a Microchip device implementing a DSU. The DSU contains identification registers to differentiate the device.

17.7.1 CoreSight Identification

A system-level Arm CoreSight ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the Arm Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:

Figure 17-4. Conceptual 64-bit Peripheral ID

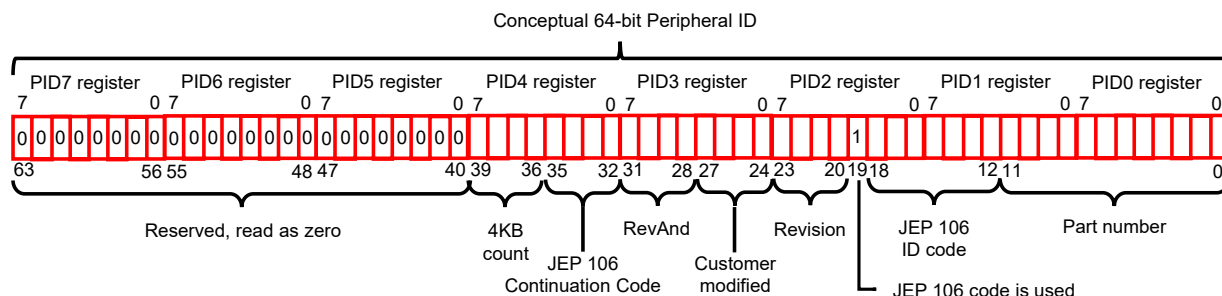


Table 17-2. Conceptual 64-Bit Peripheral ID Bit Descriptions

Field	Size	Description	Location
JEP-106 CC code	4	Microchip Continuation code: 0x0	PID4
JEP-106 ID code	7	Device ID: 0x29	PID1+PID2
4KB count	4	Indicates that the CoreSight component is a ROM: 0x0	PID4
Rev And	4	Not used; read as 0	PID3
CUSMOD	4	Not used; read as 0	PID3
PARTNUM	12	Contains 0xCD0 to indicate that DSU is present. this indicates that device identification can be completed by reading the Device Identification register (DID).	PID0+PID1
REVISION	4	DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants.	PID2

For additional information, refer to the “Arm Debug Interface Version 5 Architecture Specification”.

17.7.2 Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Chip Revision
- Part Number

17.8 Peripheral Dependencies

Table 17-3. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
DSU	0x44000000		AHB: CLKMSK[0] MASK0	---	0 STATUS0 PERID0	1 : DCC0 2 : DCC1		PD_CORE_SW

17.9 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0						PRIV		
0x04	CTRLB	31:24								
		23:16								
		15:8								
		7:0							DCCDMALVL1	DCCDMALVL0
0x08	WPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPLCK	WPEN
0x0C ... 0xFF	Reserved									
0x0100	CTRLC	31:24								
		23:16								
		15:8								
		7:0								DBGWRW
0x0104	STATUSA	31:24								
		23:16								BREXT
		15:8								CRSTEXT
		7:0	ROMCRC[1:0]				PERR	BERR		DONE
0x0108	STATUSB	31:24								
		23:16								
		15:8						HPS		DBGPRES
		7:0					DCCD1	DCCD0	BCCD1	BCCD0
0x010C ... 0x010F	Reserved									
0x0110	BCC0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x0114	BCC1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x0118	DCC0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x011C	DCC1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x0120	DID	31:24	VER[3:0]				PNMID[7:4]			
		23:16	PNMID[3:0]				PNDID[7:4]			
		15:8	PNDID[3:0]				MANID[10:7]			
		7:0	MANID[6:0]							FV
0x0124	DAL	31:24								
		23:16								
		15:8								
		7:0	CPU3[1:0]		CPU2[1:0]		CPU1[1:0]		CPU0[1:0]	
0x0128 ... 0x0FFF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1000	ENTRY0	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1004	ENTRY1	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1008	ENTRY2	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x100C	ENTRY3	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1010	ENTRY4	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1014	ENTRY5	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1018	ENTRY6	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x101C	ENTRY7	31:24	ADDOFF[19:12]							
		23:16	ADDOFF[11:4]							
		15:8	ADDOFF[3:0]							
		7:0							FMT	EPRES
0x1020 ... 0x1FCB	Reserved									
0x1FCC	MEMTYPE	31:24								
		23:16								
		15:8								
		7:0								SYSTEM
0x1FD0	PID4	31:24								
		23:16								
		15:8								
		7:0	SIZE[3:0]				DES2[3:0]			
0x1FD4	PID5	31:24								
		23:16								
		15:8								
		7:0	RES0[7:0]							
0x1FD8	PID6	31:24								
		23:16								
		15:8								
		7:0	RES0[7:0]							
0x1FDC	PID7	31:24								
		23:16								
		15:8								
		7:0	RES0[7:0]							
0x1FE0	PID0	31:24								
		23:16								
		15:8								
		7:0	PART0[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x1FE4	PID1	31:24								
		23:16								
		15:8								
		7:0	DES0[3:0]				PART1[3:0]			
0x1FE8	PID2	31:24								
		23:16								
		15:8								
		7:0	REVISION[3:0]				JEDEC	DES1[2:0]		
0x1FEC	PID3	31:24								
		23:16								
		15:8								
		7:0	REVAND[3:0]				CMOD[3:0]			
0x1FF0	CID0	31:24								
		23:16								
		15:8								
		7:0	PRMBL0[7:0]							
0x1FF4	CID1	31:24								
		23:16								
		15:8								
		7:0	CCLASS[3:0]				PRMBL1[3:0]			
0x1FF8	CID2	31:24								
		23:16								
		15:8								
		7:0	PRMBL2[7:0]							
0x1FFC	CID3	31:24								
		23:16								
		15:8								
		7:0	PRMBL3[7:0]							

17.9.1 Control A REGISTER

Name: CTRLA
Offset: 0x0
Reset: 0x00000000
Property: R/W

Table 17-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						PRIV		
Access						R/W		
Reset						0		

Bit 2 – PRIV DAP AHB access to DSU Internal Address Space registers are also affected by PRIV setting. If PRIV=1, non-privileged DAP AHB accesses to DSU Internal Address Space registers return an AHB slave bus error. DSU External address space and DSU CoreSight ROM Address Space are not affected by the PRIV bit.

DAPAHB access to DSU Internal Address Space registers are also affected by PRIV setting. If PRIV=1, non-privileged DAP AHB accesses to DSU Internal Address Space registers return an AHB Client bus error. DSU External address space and DSU CoreSight ROM Address Space are not affected by the PRIV bit.

Value	Name	Description
1		Internal Address Space registers only accessible in privileged mode. Test functions generate privileged requests.
0		Internal Address Space registers accessible in privileged and unprivileged modes. Test functions generate unprivileged requests.

17.9.2 Control B REGISTER

Name: CTRLB
Offset: 0x4
Reset: 0x00000000
Property: R/W

Table 17-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DCCDMALVL	DCCDMALVL
							1	0
Access							R/W	R/W
Reset							0	0

Bits 0, 1 – DCCDMALVL DMA Trigger n Level

Value	Name	Description
1	FULL	Trigger n rises when DCCn is written and falls when it is read
0	EMPTY	Trigger n rises when DCCn is read and falls when it is written

17.9.3 Write Protection Control REGISTER

Name: WPCTRL
Offset: 0x8
Reset: 0x00000000
Property: R/W

Table 17-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPLCK	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Writing a value other than DSU_WPCTRL_KEY to this field cancels the write operation to this register and generates a slave bus error. This field always returns 0 on read.

Bit 1 – WPLCK

Value	Name	Description
1		WPCTRL register is write protected.
0		WPCTRL is not write protected.

Bit 0 – WPEN Write Protection Enable

Value	Name	Description
1		Register write protection enabled. Non-debugger writes to registers marked with Write-Protect property are canceled and generate a slave bus error.
0		Register write protection disabled.

17.9.4 Control C REGISTER

Name: CTRLC
Offset: 0x100
Reset: 0x00000000
Property: R/W

Table 17-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								DBGRW
Access								R/W
Reset								0

Bit 0 – DBGRW If ext_cpu_me[n]=0, this bit field becomes undefined where reads return 0 and writes are ignored. This does not apply to DBGRW0 since CPU0 can never be disabled by ext_cpu_me[0].

Value	Name	Description
1		Debugger accesses through CPU _n do not carry the debug attribute (i.e debugger reads are destructive).
0		Allow debugger accesses through CPU _n to carry the debug attribute (i.e debugger reads are non-destructive).

17.9.5 Status A REGISTER

Name: STATUSA
Offset: 0x104
Reset: 0x00000000
Property: R/K

Table 17-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								BREXT
Access								HS/HC/R/K
Reset								0
Bit	15	14	13	12	11	10	9	8
								CRSTEXT
Access								HS/HC/R/K
Reset								0
Bit	7	6	5	4	3	2	1	0
	ROMCRC[1:0]				PERR	BERR		DONE
Access	HS/HC/R		HS/HC/R		HS/R/K	HS/R/K		HS/R/K
Reset	0		0		0	0		0

Bit 16 – BREXT If ext_cpu_me[n]=0, this bit field becomes undefined where reads return 0 and writes are ignored. This does not apply to BREXT0 since CPU0 can never be disabled by ext_cpu_me[0].

Value	Name	Description
1		Boot ROM Phase Extension for CPUx. Set on Cold-Plugging detection.
0		Boot ROM Phase Extension released for CPUx.

Bit 8 – CRSTEXT If ext_cpu_me[n]=0, this bit field becomes undefined where reads return 0 and writes are ignored. This does not apply to CRSTEXT0 since CPU0 can never be disabled by ext_cpu_me[0].

Value	Name	Description
1		CPUx held in reset extension. Set on Cold-Plugging detection.
0		CPUx released from reset extension.

Bits 7:6 – ROMCRC[1:0] Boot ROM CRC Self-Check Status

00 = CRC is in progress.

10 = CRC completed and passed.

01 = CRC completed but failed indicating possible issue with the ROM content. System held in lockup state.

11 = CRC never executed because user disabled CRC check through fuse settings.

Value	Name	Description
0		CRC is in progress
2		CRC completed and passed
1		CRC completed but failed indicating possible issue with the ROM content. System held in lockup state.
3		CRC never executed because user disabled CRC check through fuse settings.

Bit 3 – PERR Protection Error Status

Value	Name	Description
1		Invalid command written to TCTRL.CMD, partial write to the upper or lower bytes of TCTRL.CMD, or invalid key written to SAFEDOOR.
0		No protection error detected

Bit 2 – BERR Bus Error Status

Value	Name	Description
1		Bus error incurred while performing the DSU command.
0		No bus error detected

Bit 0 – DONE Done Status

Value	Name	Description
1		TCTRL.CMD command completed
0		DSU is executing a command or idle.

17.9.6 Status B REGISTER

Name: STATUSB
Offset: 0x108
Reset: 0x00000000
Property: R

Table 17-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						HPS		DBGPRES
Access						HC/HS/R		HC/HS/R
Reset						0		0
Bit	7	6	5	4	3	2	1	0
					DCCD1	DCCD0	BCCD1	BCCD0
Access					HC/HS/R	HC/HS/R	HC/HS/R	HC/HS/R
Reset					0	0	0	0

Bit 10 – HPS Hot-Plugging Status

1 = Hot-Plugging active
0 = Hot-Plugging inactive

Value	Name	Description
1		Hot-Plugging active
0		Hot-Plugging inactive

Bit 8 – DBGPRES Debugger Present Status

This bit is set when a debugger probe is detected.
Only a POR or external reset can reset this bit.

Value	Name	Description
1		Debugger probe detected
0		No debugger detected

Bits 2, 3 – DCCD Debug Communication Channel x Dirty

Value	Name	Description
1		DCCDx.DATA register was written to.
0		DCCDx.DATA register was read from.

Bits 0, 1 – BCCD Boot ROM Communication Channel x Dirty

Value	Name	Description
1		BCCDx.DATA register was written to.
0		BCCDx.DATA register was read from.

17.9.7 Boot ROM Channel n REGISTER

Name: BCC
Offset: 0x0110 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: R/W

Table 17-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data register used for ROM commands/responses. Writing to the register sets STATUCB.BCCDn. Reading from the register clears STATUSB.BCCDn.

Writing BCCx sets STATUSB.BCCx.

Reading BCCx clears STATUSB.BCCx.

Notes:

1. BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.

17.9.8 Debug Communication Channel n REGISTER

Name: DCC
Offset: 0x0118 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: R/W

Table 17-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data register used for debug communication. Writing to the register sets STATUCB.DCCDn. Reading from the register clears STATUSB.DCCDn.

Writing DCCx sets STATUSB.DCCx.

Reading DCCx clears STATUSB.DCCx.

Notes:

1. BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.

17.9.9 Device Identification Register

Name: DID
Offset: 0x120
Reset: 0x00000053
Property: R/W

Table 17-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	VER[3:0]				PNMID[7:4]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PNMID[3:0]				PNDID[7:4]			
Access	R	R	R	R	P/R/W	P/R/W	P/R/W	P/R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PNDID[3:0]				MANID[10:7]			
Access	P/R/W	P/R/W	P/R/W	P/R/W	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MANID[6:0]							FV
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	1	0	0	1	1

Bits 31:28 – VER[3:0] Provides the version code where a value of 0 represents A.0

Bits 27:20 – PNMID[7:0] Provides bits [15:8] of the Device Part Number, where the entire Part Number is given by PN[15:0] = DID[27:12]

Bits 19:12 – PNDID[7:0] Provides bits [7:0] of the Device Part Number, where the entire Part Number is given by PN[15:0] = DID[27:12] This register field is loaded by fuse configuration and is writable in test mode (i.e. when TESTMODE0.ENABLE=1).

Bits 11:1 – MANID[10:0] This field contains the value defined in JEDEC Publication JEP106Q for Microchip Technology Inc., 0x29.

This is duplicate of the JEP-106 CC and ID code present in the Device Service Unit Coresight ROM table that always read 0x29.

Bit 0 – FV Fixed Value of 1

17.9.10 Debugger Access Level REGISTER

Name: DAL
Offset: 0x124
Reset: 0x00000000
Property: R

Table 17-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	CPU3[1:0]		CPU2[1:0]		CPU1[1:0]		CPU0[1:0]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 2:3, 4:5, 6:7 – CPU Condition: {DSU_TZ_IMPLEMENTED==0} 01 = Reserved

Value	Name	Description
3	UNIMPLEMENTED	Unimplemented CPU
2	FULL_DEBUG	Debugger has full access to debug CPU
1	NS_DEBUG	Debugger can access only non-secure regions
0	SECURED	Debugger access is disabled
3	UNIMPLEMENTED	Unimplemented CPU
2	FULL_DEBUG	Debugger has full access to debug CPU
0	SECURED	Debugger access is disabled

Bits 1:0 – CPU0[1:0] Condition: {DSU_TZ_IMPLEMENTED==1} 11 = Reserved Condition: {DSU_TZ_IMPLEMENTED==0} 11 = Reserved 01 = Reserved

Writing in this bitfield has no effect.

0x0 = (SECURED) Debugger targeting CPU0 domain can only access the DSU external address space otherwise debugger access is disabled

0x1 = (NS_DEBUG) Debugger can access only non-secure regions. This value is only applicable to PIC32CM SG devices.

0x2 = (FULL_DEBUG) Debugger can access secure and non-secure regions.

0x3 = Reserved.

Value	Name	Description
2	FULL_DEBUG	Debugger can access secure and non-secure regions
1	NS_DEBUG	Debugger can access only non-secure regions
0	SECURED	Debugger targeting CPU0 domain can only access the DSU external address space otherwise debugger access is disabled
2	FULL_DEBUG	Debugger has unrestricted access.
0	SECURED	Debugger targeting CPU0 domain can only access the DSU external address space otherwise debugger access is disabled

17.9.11 CoreSight ROM Table Entry N Register

Name: ENTRY0
Offset: 0x1000
Reset: 0x00000002
Property: R

Table 17-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R/HS/HC
Reset							1	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES This bit is set to 0 by hardware under the following conditions $DAL.CPU0 == DAL0$ or $n > DSU\_CPU\_NUM$, meaning the CPU interface does not exist, or $ext\_cpu\_me[n] == 0$ when $n \leq DSU\_CPU\_NUM$, meaning the CPU exists but is disabled otherwise this bit is set to $DSU\_CORESIGHT\_ENTRY[n][0]$.

Value	Name	Description
1		Entry present
0		Entry not present

17.9.12 CoreSight ROM Table Entry N Register

Name: ENTRY1
Offset: 0x1004
Reset: 0x00000002
Property: R

Table 17-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R/HS/HC
Reset							1	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES This bit is set to 0 by hardware under the following conditions $DAL.CPU0 == DAL0$ or $n > DSU\_CPU\_NUM$, meaning the CPU interface does not exist, or $ext\_cpu\_me[n] == 0$ when $n \leq DSU\_CPU\_NUM$, meaning the CPU exists but is disabled otherwise this bit is set to $DSU\_CORESIGHT\_ENTRY[n][0]$.

Value	Name	Description
1		Entry present
0		Entry not present

17.9.13 CoreSight ROM Table Entry N Register

Name: ENTRY2
Offset: 0x1008
Reset: 0x00000002
Property: R

Table 17-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R/HS/HC
Reset							1	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES This bit is set to 0 by hardware under the following conditions $DAL.CPU0 == DAL0$ or $n > DSU\_CPU\_NUM$, meaning the CPU interface does not exist, or $ext\_cpu\_me[n] == 0$ when $n \leq DSU\_CPU\_NUM$, meaning the CPU exists but is disabled otherwise this bit is set to $DSU\_CORESIGHT\_ENTRY[n][0]$.

Value	Name	Description
1		Entry present
0		Entry not present

17.9.14 CoreSight ROM Table Entry N Register

Name: ENTRY3
Offset: 0x100C
Reset: 0x00000002
Property: R

Table 17-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R/HS/HC
Reset							1	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES This bit is set to 0 by hardware under the following conditions $DAL\_CPU0 == DAL0$ or $n > DSU\_CPU\_NUM$, meaning the CPU interface does not exist, or $ext\_cpu\_me[n] == 0$ when $n \leq DSU\_CPU\_NUM$, meaning the CPU exists but is disabled otherwise this bit is set to $DSU\_CORESIGHT\_ENTRY[n][0]$.

Value	Name	Description
1		Entry present
0		Entry not present

17.9.15 CoreSight ROM Table Entry 4 Register

Name: ENTRY4
Offset: 0x1010
Reset: 0x00000000
Property: R

Table 17-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							0	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES CoreSight Entry Present

Value	Name	Description
1		Entry present
0		Entry not present

17.9.16 CoreSight ROM Table Entry 5 Register

Name: ENTRY5
Offset: 0x1014
Reset: 0x00000000
Property: R

Table 17-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							0	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES CoreSight Entry Present

Value	Name	Description
1		Entry present
0		Entry not present

17.9.17 CoreSight ROM Table Entry 6 Register

Name: ENTRY6
Offset: 0x1018
Reset: 0x00000000
Property: R

Table 17-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							0	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES CoreSight Entry Present

Value	Name	Description
1		Entry present
0		Entry not present

17.9.18 CoreSight ROM Table Entry 7 Register

Name: ENTRY7
Offset: 0x101C
Reset: 0x00000000
Property: R

Table 17-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDOFF[19:12]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDOFF[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDOFF[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
							FMT	EPRES
Access							R	R
Reset							0	0

Bits 31:12 – ADDOFF[19:0] Base address of the component, relative to the base address of this ROM Table.

Bit 1 – FMT CoreSight Rom Table Format

Value	Name	Description
1		32-bit format
0		8-bit format

Bit 0 – EPRES CoreSight Entry Present

Value	Name	Description
1		Entry present
0		Entry not present

17.9.19 CoreSight ROM Table Memory Type Register

Name: MEMTYPE
Offset: 0x1fcc
Reset: 0x00000000
Property: R

Table 17-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								SYSTEM
Access								HS/HC/R
Reset								0

Bit 0 – SYSMEM Value of this register field is set to 1 when CPU0 effective DAL is DAL1 or DAL2 otherwise set to 0.

This bit indicates whether system memory is present on the bus that connects to the ROM table. This bit is set at power-up when DAL.CPU0 is greater than 0 indicating that the system memory is accessible from a debug adapter.

This bit is cleared at power-up when DAL.CPU0 is equal to 0 indicating that the system memory is not accessible from a debug adapter.

Value	Name	Description
1		System memory is also present on this bus
0		System memory not present on bus. This is a dedicated debug bus.

17.9.20 CoreSight Peripheral Identification 4 Register

Name: PID4
Offset: 0x1fd0
Reset: 0x00000000
Property: R

Table 17-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	SIZE[3:0]				DES2[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – SIZE[3:0] Always returns 0x0, which indicates this component uses a total contiguous size of memory window of 4KB.

These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

Bits 3:0 – DES2[3:0] Always returns 0x0

These bits will always return zero when read, indicating a Microchip device.

17.9.21 CoreSight Peripheral Identification 5 Register

Name: PID5
Offset: 0x1fd4
Reset: 0x00000000
Property: R

The bits in this register are unimplemented and always read as '0'.

Table 17-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	RES0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RES0[7:0] Always returns 0x00

17.9.22 CoreSight Peripheral Identification 6 Register

Name: PID6
Offset: 0x1fd8
Reset: 0x00000000
Property: R

The bits in this register are unimplemented and always read as '0'.

Table 17-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	RES0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RES0[7:0] Always returns 0x00

17.9.23 CoreSight Peripheral Identification 7 Register

Name: PID7
Offset: 0x1fdc
Reset: 0x00000000
Property: R

The bits in this register are unimplemented and always read as '0'.

Table 17-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	RES0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RES0[7:0] Always returns 0x00

17.9.24 CoreSight Peripheral Identification 0 Register

Name: PID0
Offset: 0x1fe0
Reset: 0x000000D0
Property: R

Table 17-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PART0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	0	1	0	0	0	0

Bits 7:0 – PART0[7:0] Always returns 0xD0 (Microchip DSU Part Number is 0xCD0)
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.

17.9.25 CoreSight Peripheral Identification 1 Register

Name: PID1
Offset: 0x1fe4
Reset: 0x0000009C
Property: R

Table 17-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	DES0[3:0]				PART1[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	1	1	1	0	0

Bits 7:4 – DES0[3:0] Always returns 0x9 for Microchip Devices (Microchip JEP106 Identity Code = 0x29)
These bits will always return 0x9 when read, indicating a Microchip device (Microchip JEP-106 identity code is 0x29).

Bits 3:0 – PART1[3:0] Always returns 0xC (Microchip DSU Part Number is 0xCD0)
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.

17.9.26 CoreSight Peripheral Identification 2 Register

Name: PID2
Offset: 0x1fe8
Reset: 0x0000000A
Property: R

Table 17-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	REVISION[3:0]				JEDEC	DES1[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	1	0

Bits 7:4 – REVISION[3:0] Provides the REVISION value to identify this particular DSU.
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

Bit 3 – JEDEC Always returns 0x1 which indicates a JEDEC value is used.
Always returns 0x1 which indicates a JEDEC value is used.

Bits 2:0 – DES1[2:0] Always returns 0x02, indicating a Microchip Device (Microchip JEP106 Identity Code = 0x29)
These bits will always return 0x2 when read, indicating a Microchip device (Microchip JEP-106 identity code is 0x29).

17.9.27 CoreSight Peripheral Identification 3 Register

Name: PID3
Offset: 0x1fec
Reset: 0x00000000
Property: R

Table 17-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	REVAND[3:0]				CMOD[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – REVAND[3:0] Always returns 0x0
These bits will always return 0x0 when read.

Bits 3:0 – CMOD[3:0] Always returns 0x0
These bits will always return 0x0 when read.

17.9.28 CoreSight Component Identification 0 Register

Name: CID0
Offset: 0x1ff0
Reset: 0x0000000D
Property: R

Table 17-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PRMBL0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	1	0	1

Bits 7:0 – PRMBL0[7:0] Always returns 0x0D
These bits will always return 0xD when read.

17.9.29 CoreSight Component Identification 1 Register

Name: CID1
Offset: 0x1ff4
Reset: 0x00000010
Property: R

Table 17-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	CCLASS[3:0]				PRMBL1[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0

Bits 7:4 – CCLASS[3:0] Always returns 0x1 which identifies this component as ARM CoreSight ROM Table. These bits will always return 0x1 when read indicating that this Arm CoreSight component is ROM table (refer to the Arm Debug Interface v5 Architecture Specification at www.arm.com).

Bits 3:0 – PRMBL1[3:0] Always returns 0x0
These bits will always return 0x00 when read.

17.9.30 CoreSight Component Identification 2 Register

Name: CID2
Offset: 0x1ff8
Reset: 0x00000005
Property: R

Table 17-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PRMBL2[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	0	1

Bits 7:0 – PRMBL2[7:0] Always returns 0x05
These bits will always return 0x5 when read.

17.9.31 CoreSight Component Identification 3 Register

Name: CID3
Offset: 0x1ffc
Reset: 0x000000B1
Property: R

Table 17-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PRMBL3[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	1	1	0	0	0	1

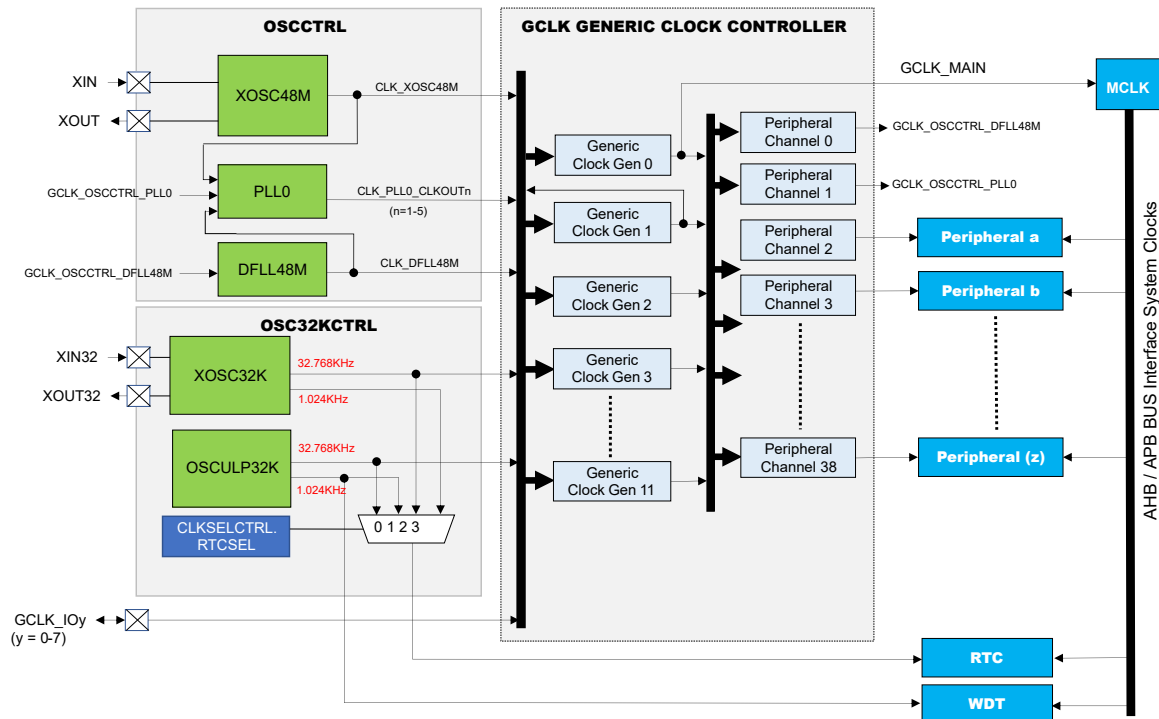
Bits 7:0 – PRMBL3[7:0] Always returns 0xB1
These bits will always return 0xB1 when read.

18. Clock Distribution System

This chapter summarizes the clock distribution and terminology in the PIC32CM SG00/GC00 family of devices. For additional information refer to the peripheral chapters and the *GCLK Generic Clock Controller* documentation.

18.1 Clock Distribution

Figure 18-1. Clock Distribution Block Diagram



18.2 Clock System Features

The PIC32CM SG00/GC00 clock system features are as follows:

- *Clock sources*, that is oscillators controlled by OSCCTRL and OSC32KCTRL:
 - A clock source provides a time base that is used by other components, such as Generic Clock Generators. Example clock sources are the internal 48 MHz DFLL48M, Ultra Low-power 32 kHz RC oscillator, OSCULP32K, PLL and external oscillators XOSC and XOSC32K.
- *Generic Clock Controller (GCLK)*, which generates, controls and distributes asynchronous clocks consisting of:
 - *Generic Clock Generators*: These have programmable prescalers that can use any of the system clock sources. The Generic Clock Generator 0 generates the clock signal GCLK_MAIN, which is used by the CPU and Data Ram Tightly Coupled Memory, which in turn generates synchronous clocks.
 - *Generic Clocks*: These are clock signals generated by Generic Clock Generators. They are the Peripheral Channels and serve as clocks for the peripherals of the system. Multiple instances of a peripheral will typically have a separate Generic Clock.
- *Main Clock Controller (MCLK)*
 - The MCLK generates and controls the synchronous clocks for the system. This includes the CPU, bus clocks (AHB (Advanced High-performance Bus), APB (Advanced Peripheral Bus) and

AXI (Advanced eXtensible Interface)), as well as the Special Function Register interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

To customize the clock distribution, refer to these registers and bit fields:

- The source oscillator for a generic clock generator 'n' is selected by writing to the Source bit field in the Generator Control n, (n=0-15), register (GCLK.GENCTRLn.SRC).
- A Peripheral Channel m can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN).
- The Peripheral Channel number, *m*, is fixed for a given peripheral. See the Mapping table in the description of GCLK.PCHCTRLm, (i.e., GCLK Chapter).
- The AHB/APB/AIX Peripheral BUS clocks are enabled and disabled by writing to the respective bit in the AHB/APB/AIX Mask register. The AHB/APB/AIX clocks are enabled and disabled by writing to the respective bit in the AHB/APB/AIX Mask register (MCLK.CLKMSK[0,1,2,3].MSKn). They are enabled by default on reset.

18.3 Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, peripheral accesses by the CPU need to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is complete.

In the data sheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

18.4 Register Synchronization

18.4.1 Overview

All peripherals are composed of one digital bus interface connected to the APB, AHB or AXI bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

All registers in the bus interface are accessible without synchronization.

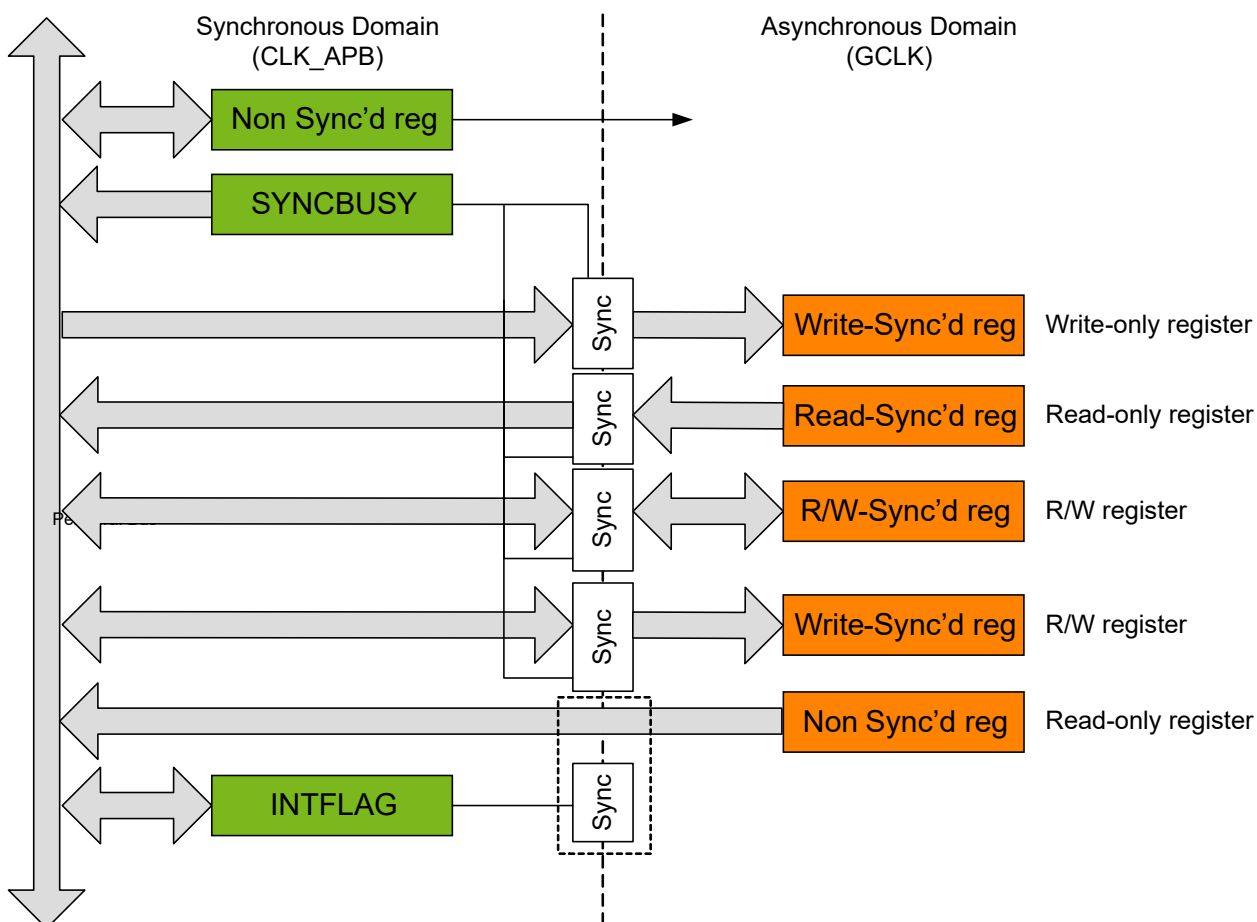
All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

Each individual register description will have the properties "Read-Synchronized" and/or "Write-Synchronized" if a register is synchronized.

As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).

Note: For registers requiring both read-synchronization and write-synchronization, the corresponding bit in the SYNCBUSY register is shared.

Figure 18-2. Register Synchronization Overview



18.4.2 General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain (GCLK). The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer to Synchronization Delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported through the Peripheral Access Controller (PAC).

Example:

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

Offset	Register
0x00	REGA
0x01	REGB
0x02	REGC
0x03	

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated through the PAC.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

18.4.3 General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.

Note: The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore, reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before Sleep mode.

In addition, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

18.4.4 Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e., when all bits in SYNCBUSY are '0'.

18.4.5 Write Synchronization for CTRLA.ENABLE

Setting the Enable bit in a peripheral module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronization Ready interrupt (if available) cannot be used to enable write-synchronization.

18.4.6 Write-Synchronization for Software Reset Bit

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a '1' to the CTRLA.SWRST bit it will immediately read as '1'.

CTRLA.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRLA.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

Note: Not all peripherals have the SWRST bit in the respective CTRLA register.

18.4.7 Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay D is within the range of:

$$5 \times P_{GCLK} + 2 \times P_{APB} < D < 6 \times P_{GCLK} + 3 \times P_{APB}$$

Where P_{GCLK} is the period of the generic clock and P_{APB} is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2 \times P_{APB}$.

18.5 Enabling a Peripheral

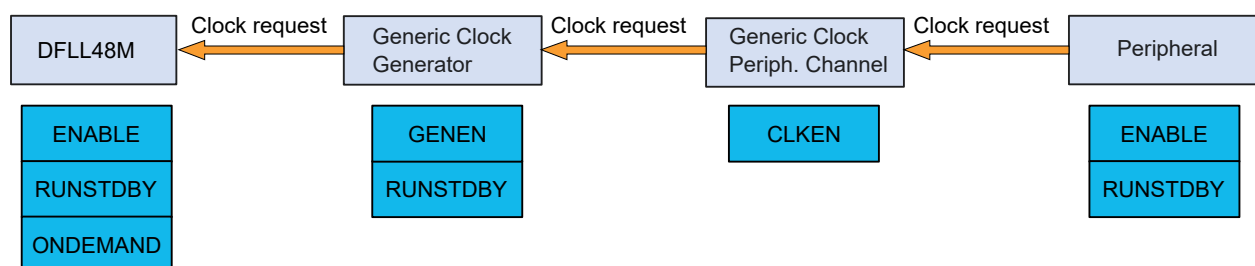
In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

- A running Clock Source as selected in the OSCCTRL Module
- A clock from the Generic Clock Generator (GCLK.GENCTRLn.SRC), must be configured to use one of the running Clock Sources, and the Generator must be enabled

- The Peripheral Channel that provides the Generic Clock signal to the designated peripheral "m", (i.e., PCHCTRLm), must be configured to use a running Generic Clock Generator (i.e., PCHCTRLm.GEN), and the Generic Clock must be enabled
- The user system bus interface of the peripherals are enabled by default on reset, (i.e., MCLK.CLKMSK[0,1].MSKn = 1), and doesn't need to be unmasked to enable it unless disabled at run time by the users software and the user needs to access the peripheral once again. In that case the user must re-enable, unmask, in the Main Clock Controller (MCLK.CLKMSK[0,1].MSKn = 1). If this is not done the peripheral in question registers will read all '0's and any writing attempts to the peripheral will be discarded.

18.6 On Demand Clock Requests

Figure 18-3. Clock Request Routing



All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral, through the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started or kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in between are enabled. The time taken from a clock request being asserted to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time T_{start} from a clock request until the clock is available for the peripheral is between:

$$T_{start\_max} = \text{Clock source startup time} + 2 \times \text{clock source periods} + 2 \times \text{divided clock source periods}$$

$$T_{start\_min} = \text{Clock source startup time} + 1 \times \text{clock source period} + 1 \times \text{divided clock source period}$$

The time between the last active clock request stopped and the clock is shut down, T_{stop} , is between:

$$T_{stop\_min} = 1 \times \text{divided clock source period} + 1 \times \text{clock source period}$$

$$T_{stop\_max} = 2 \times \text{divided clock source periods} + 2 \times \text{clock source periods}$$

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules.

18.7 Power Consumption Versus Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

Clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent

on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

18.8 Clocks after Reset

On any Reset the synchronous clocks start to their initial state:

- DFLL48M is enabled and configured to run at 48 MHz Open Loop
- Generic Clock Generator 0 uses DFLL48M by default as a source and generates GCLK_MAIN
- CPU and BUS clocks are undivided and enabled

On a Power-on Reset, the 32 kHz clock sources are reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except Generator 0
- All Peripheral Channels in GCLK are disabled

On a User Reset the GCLK module starts to its initial state, except for:

- Generic Clocks that are write-locked, for example, the according WRTLOCK is set to 1 prior to Reset

19. Oscillator Controller (OSCCTRL)

19.1 Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC and oscillator used during start-up operation (DFLL). Each oscillator is controlled by dedicated registers in the OSCCTRL. Through this interface, the oscillators are enabled, disabled, or have their configuration values updated. The Status register gathers different status signals coming from the oscillators controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from Sleep mode, provided the corresponding interrupt is enabled. Generated events can be used by the system to trigger actions on other peripherals

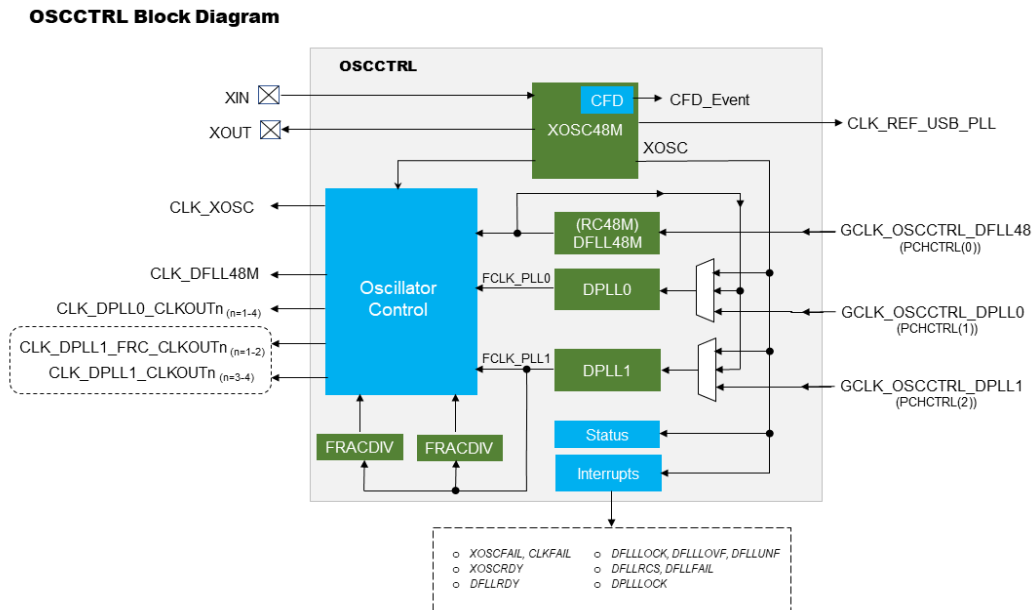
19.2 Features

The following are key features of the OSCCTRL module:

- 48 MHz Crystal Oscillator (XOSC48)
 - Supports 4 MHz to 48 MHz crystal oscillators
 - Supports ceramic resonators up to 10 MHz
 - AGC, Automatic gain control loop with manual override
 - User clock Ready Status
 - Clock-fail detection in AGC mode with safe clock switch
- RC48M (DFLL48M)
 - System clock output
 - 48.0 MHz (-5% to 2%) calibrated with DFLL48M in open-loop operation
 - 48.0 MHz ($\pm 0.25\%$) calibrated with DFLL48M in closed-loop operation
 - 8 MHz ($\pm 15\%$) in low-power mode with DFLL48M in open-loop operation
- One PLL, Phase-Locked Loop
 - 4 MHz to 48 MHz Reference Input Clock
 - Programmable Reference Input Clock divider
 - Single-ended output frequency: 12.7 MHz to 200 MHz or 1.6 GHz for fractional divider module input
 - Six single-ended outputs with programmable output clock dividers
 - Programmable closed loop bandwidth

19.3 OSCCTRL Block Diagram

Figure 19-1. OSCCTRL Block Diagram



19.4 Signal Descriptions

Signal Name	Type	Description
X _{IN}	Analog Input	Primary Multipurpose Crystal Oscillator or external clock generator input
X _{OUT}	Analog Output	Primary Multipurpose Crystal Oscillator output

19.5 Functional Description

19.5.1 Principles of Operation

The XOSC, DFLL48M, and PLL0 are configured through the OSCCTRL control registers. Through this interface, the oscillators are enabled, disabled, or have their calibration values updated. The Status register gathers different status signals coming from the oscillators controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from Sleep mode, provided the corresponding interrupt is enabled.

19.5.2 External Multipurpose Crystal Oscillator (XOSC) Operation

The XOSC can operate in the following two modes:

- External clock with an external clock signal connected to the XIN pin.
- Crystal oscillator with an external 4-48 MHz crystal.

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pins will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRLA.ENABLE). To enable XOSC as an external crystal oscillator, the XTAL Enable bit (XOSCCTRLA.XTALEN) must be written to '1'. If XOSCCTRLA.XTALEN is zero, the external clock input on XIN will be enabled.

If the External Multipurpose Crystal Oscillator AGC, Auto Gain Control, Loop (XOSCCTRLA.AGC) is '1', the oscillator gain will be automatically adjusted starting from lowest setting, and will increase the gain in sequential steps every XOSCCTRLB.GBW[1:0] delay for a maximum of 16 steps accordingly, until a valid stable oscillation is detected which will result in lowest power consumption for a stable crystal oscillation. In this mode, XOSCCTRLA.AGC = 1, the manual XOSC crystal gain settings XOSCCTRLB.GMAN[1:0] are ignored.

The AGC Gain control loop update rate can be controlled by the XOSCCTRLB.GBW[1:0] user configuration bits. This controls the AGC delays between each gain step increase. Some unique crystals may have slow start-up times, so to insure the lowest power gain setting it may be necessary to increase the AGC gain step delay update rate to allow enough time for a slow crystal start-up to be realized and detected before increasing the XOSC gain step again. This is rare and 95% of all 4 MHz to 48 MHz crystals start-up with the default XOSCCTRLB.GBW[1:0] value.

Alternatively, if the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is '0', the user can manually select the crystal oscillator operating condition by setting the manual gain value in the XOSCCTRLB.GMAN[1:0] register. In this mode, XOSCCTRLA.AGC = 0, XOSCCTRLB.GBW[1:0] AGC update rate is ignored.

Start-Up time, XOSCCTRLA.STARTUP, select the maximum start-up time for the oscillator XOSC before a clock fail is acknowledged. The OSCULP32K oscillator is used to clock the start-up counter for the XOSC. Start-Up Time.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRLA.ONDEMAND, and XOSCCTRLA.ENABLE. XOSCCTRLA.ONDEMAND must be written when XOSCCTRLA.ENABLE = 0. Otherwise, the write of this bit is ignored. If XOSCCTRLA.ENABLE = 0, the XOSC will be always stopped. For XOSCCTRLA.ENABLE = 1, this table is valid:

Table 19-1. XOSC Sleep Behavior

CPU Mode	ON DEMAND	Sleep Behavior of XOSC and CFD
Active or Idle	0	Always run
Active or Idle	1	Run if requested by a peripheral
Standby	0	Always run
Standby	1	Run if requested by a peripheral
Backup	0	Always OFF
Backup	1	Always OFF

After a hard reset, or when waking up from a Sleep mode where the XOSC was disabled, the XOSC will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRLA.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic. The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set when the external clock or crystal oscillator is stable and ready to be used as a clock source. The INTFLAG.XOSCRDY bit is set on a zero-to-one transition of STATUS.XOSCRDY and an interrupt is generated if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set. If the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is set, and the XOSC fail to oscillate after attempting all gain settings, the External Multipurpose Crystal Oscillator Startup Fail bit in the Status register (STATUS.XOSCFAIL) is set. The INTFLAG.XOSCFAIL bit is set on a zero-to-one transition of STATUS.XOSCFAIL and an interrupt is generated if the XOSC Startup Failure bit in the Interrupt Enable Set register (INTENSET.XOSCFAIL) is set. After the the

startup time has elapsed and the External Multipurpose Crystal Oscillator did not fail oscillating, the output clock can also be monitored for failure by using the Clock Fail Detector (CFD). Refer to next section for CFD operations.

USBHS Reference Clock Division

The XOSC oscillator is the source of the USBHS PLL's reference clocks. Each USBHS PLL can request the XOSC as a reference clock. Upon request by a USBHS, the XOSC clock is prescaled by a clock divider and issued to the USBHS PLL. The clock division ratio and enable is configured by the XOSCCTRLA.USBHSDIV bitfield.

19.5.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator clock signal provided by the External Multipurpose Crystal Oscillator (XOSC). It detects failing operation of the XOSC clock and allows to switch to a safe clock in case of clock failure. The safe clock is derived from the DFLL48M with a configurable prescaler. The user can also switch from the safe clock to the XOSC clock in case of clock recovery. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller. The CFD operation is automatically suspended when the XOSC clock is not requested in ONDEMAND mode or halted in STANDBY.

The user interface registers allow to enable, disable and configure the CFD. The Status register gives status on failure and clock switch conditions. The Clock Failure Detector can optionally trigger an interrupt or an event when a failure is detected.

Note: If the measured clock is intermittent or too slow, the STATUS:XOSCFAIL flag may appear to be toggling on and off until the clock is stopped or restored. However, the first rising edge will in any case be captured and logged in INTFLAG:XOSCFAIL.

Clock Failure Detection

At reset, the CFD is disabled. The CFD does not monitor the XOSC clock when the oscillator is disabled (XOSCCTRLA.ENABLE = 0).

Before starting the CFD operation, the user must start and enable the safe clock source (DFLL48M). To start the CFD operation, the user must write a one to the CFD Enable bit in the External Oscillator Control register (XOSCCTRLA.CFDEN). After the start or restart of the XOSC, the CFD does not detect failure until the start-up time, as configured by the Oscillator Start-Up Time (XOSCCTRLA.STARTUP) in the External Multipurpose Crystal Oscillator Control register, is elapsed. Once the XOSC Start-Up Time is elapsed, the XOSC clock is constantly monitored.

During a period of 4 safe clocks, the CFD watches for a clock activity from the XOSC. There must be one rising and one falling XOSC clock edges during a 4 safe clock periods to meet a non-failure status. If no activity is detected, the failure status is asserted. The Clock Failure status bit in the Status register (STATUS.CLKFAIL) is set. The Clock Failure interrupt flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) is set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) is set, an interrupt is generated. An output event is generated as well, if the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set.

The XOSC clock continues to be monitored after a clock failure. The Clock Failure status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC clock activity.

Clock Switch Back

When a clock failure is detected, the XOSC clock is replaced by the safe clock in order to maintain an active clock during the XOSC clock failure. The safe clock source can be downscaled with a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC clock is switched to the safe clock, the Clock Switch bit (STATUS.XOSCCKSW) in the Status register is set.

When the XOSC clock is switched to the safe clock, the External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set when the safe clock is stable and ready to be used as a clock source.

When the CFD has switched to the safe clock, the XOSC is not disabled. The application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations.

In the case the application can recover the XOSC, the Clock Failure status bit in the Status register (STATUS.CLKFAIL) is cleared. The application can switch back to the XOSC clock by writing a one to Switch Back bit (XOSCCTRLA.SWBEN) in the External Oscillator Control register. Once the XOSC clock is switched back, the Switch Back bit (XOSCCTRLA.SWBEN) is cleared by the hardware.

Note: In the situation that the user has wrote a one to the Switch Back bit (XOSCCTRLA.SWBEN), the APB clock is permanently requested until the switch is completed.

If the switch back can not complete due to a persistent failure, it will prevent the system from entering into low power sleep mode.

To stop the clock request the user can disable the XOSC (XOSCCTRLA.ENABLE=0) and disable the CFD detection (XOSCCTRLA.CFDEN=0).

Prescaler:

The CFD has an internal configurable prescaler (XOSCCTRLA.CFDPRESC) to generate the safe clock from the DFLL48M clock. The prescaler size allows to scale down the DFLL48M clock such that the safe clock is not higher than the XOSC clock frequency monitored by the CFD. The frequency divider is $2^{CFDPRESC}$ where CFDPRESC range from 0 to 15.

Example: for an external crystal oscillator at 8 MHz and the DFLL48M internal oscillator configured to generate a 48 MHz clock, the prescaler must select a downscale value above 6 ($48/8$), for example, 8, therefore CFDPRESC = 3.

Event:

If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

Sleep Mode:

The CFD is halted depending on configuration of the XOSC and the peripheral clock request. For further details, refer to the *Sleep Behavior table*. The CFD interrupt can be used to wake up the device from sleep modes.

Notes:

1. PLL0 must be stepped down in ≤ 75 MHz increments to ≤ 75 MHz output when entering sleep modes.
2. PLL0 must be stepped up to the operating frequency in ≤ 75 MHz increments after exiting sleep modes.
3. The step delay for both of these processes needs to be ≥ 1 us.

19.5.4 Digital Frequency Locked Loop (DFLL48M) Operation

The DFLL48M can operate in both open-loop mode and closed-loop mode. In closed-loop mode, a low-frequency clock with high accuracy should be used as the reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M can be used as a source for the generic clock generators.

19.5.4.1 Basic DFLL48M Operation DFLL48M Operating modes

The DFLL48M will behave differently in different sleep modes, based on the settings of DFLLCTRLA.ONDEMAND, and DFLLCTRLA.ENABLE. DFLLCTRLA.ONDEMAND must be written when DFLLCTRLA.ENABLE = 0 and DFLLSYNC.ENABLE = 0. Otherwise, the write of this bit is ignored. If DFLLCTRLA.ENABLE = 0, the DFLL48M will be always stopped. For DFLLCTRLA.ENABLE = 1, this table is valid:

Table 19-2. DFLL48M Sleep Behavior

CPU Mode	ON DEMAND	Sleep Behavior of DFLL48M
Active or Idle	0	Always run
Active or Idle	1	Run if requested by a peripheral
Standby	0	Always run
Standby	1	Run if requested by a peripheral
Backup	0	Always OFF
Backup	1	Always OFF

The DFLL48M is used as a clock source for the generic clock generators, as described in the GCLK chapter.

The DFLL48M is factory-calibrated for 48MHz. The frequency calibration is applied at reset.

DFLL48M Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M will be determined by the values written to the DFLL TUNE register (DFLLTUNE.TUNE).

It is possible to change the values of DFLLTUNE.TUNE and thereby the output frequency of the DFLL48M output clock, CLK_DFLL48M, while the DFLL48M is enabled and in use. CLK_DFLL48M is ready to be used when STATUS.DFLLRDY is set after enabling the DFLL48M.

DFLL48M Closed-Loop Operation

In closed-loop operation, the output frequency is continuously regulated against a reference clock. Once the multiplication factor is set, the oscillator tuning is automatically adjusted. The DFLL48M must be correctly configured before closed-loop operation can be enabled. After enabling the DFLL48M, it must be configured in the following way:

1. Enable and select a reference clock (CLK_DFLL48M_REF). CLK_DFLL48M_REF is Generic Clock Channel 0 (DFLL48M_Reference). Refer to GCLK for details.
2. Select the maximum step size allowed in finding the TUNE values by writing the appropriate values to the DFLL maximum step bit group (DFLLMUL.STEP) in the DFLL STEP register. A small step size will ensure low overshoot on the output frequency but will typically result in longer lock times. A high value might give a large overshoot but will typically provide faster locking. DFLLMUL.STEP should not be higher than 50% of the maximum value of DFLLTUNE.TUNE.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register. Care must be taken when choosing DFLLMUL.MUL so that the output frequency does not exceed the maximum frequency of the device. If the target frequency is below the minimum or above the maximum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum or maximum frequency. Write the DFLLMUL.MUL to restore the TUNE register (DFLLTUNE.TUNE) to its reset value.
4. Start the closed-Loop mode by writing a one to the DFLL Loop Enable bit (DFLLCTRLB.LOOPEN) in the DFLL Control register.

The frequency of CLK_DFLL48M (Fclkdfll48m) is given by:

Equation 19-1. FCLKDFLL48M

$$F_{CLKDFLL48M} = (DFLLMUL.MUL * F_{CLKDFLL48MREF})$$

Where.

$F_{CLKDFLL48MREF}$ is the frequency of the reference clock (CLK_DFLL48M_REF).

DFLLTUNE register is read-only in closed-loop mode and is controlled by the frequency tuner to meet the user specified frequency.

DFLL48M Frequency Locking

In the lock of the frequency search in closed-loop mode, the control logic tunes the value in DFLLTUNE.TUNE so that the output frequency is very close to the desired frequency. On lock, the DFLL Locked bit (STATUS.DFLLLOCK) in the status register will be set.

Interrupts are generated by the rising of STATUS.DFLLLOCK if INTENSET.DFLLLOCK is written to '1'.

CLK_DFLL48M is ready to be used when the DFLL Ready bit (STATUS.DFLLRDY) in the Status register is set, but the accuracy of the output frequency will not be met until the Lock state is reached. For lock times, refer to the *Electrical Characteristics* section.

DFLL48M Frequency Error Measurement

The ratio between CLK_DFLL48M_REF and CLK48M_DFLL is measured automatically when the DFLL48M is in closed-loop mode. The difference between this ratio and the value in DFLLMUL.MUL is stored in the DFLL Multiplication Ratio Difference bit group (DFLLDIFF.DIFF) in the DFLL DIFF register. The relative error on CLK_DFLL48M compared to the target frequency is calculated as follows:

$$ERROR = (DIFF / MUL)$$

DFLL48M Drift Compensation

If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is zero, the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks. This means that DFLLTUNE.TUNE can change after every measurement of CLK_DFLL48M_REF. If the DFLLTUNE.TUNE value overflows or underflows due to large drift in temperature and/or voltage, the DFLL overflow or underflow bits (STATUS.DFLLLOVF or STATUS.DFLLUNF) in the Status register will be set. After an overflow or underflow error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency. An interrupt is generated on a zero-to-one transition on STATUS.DFLLLOVF or STATUS.DFLLUNF if the DFLL overflow or underflow bit (INTENSET.DFLLLOVF or STATUS.DFLLUNF) in the Interrupt Enable Set register is set. If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is one, the DFLLTUNE.TUNE values will stay constant after the lock. The user can check for a possible drift by reading the frequency error in the DFLL Multiplication Ratio Difference bit group (DFLLDIFF.DIFF).

DFLL48M Reference Clock Stop Detection

If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2¹⁷)), the DFLL Reference Clock Stopped bit (STATUS.DFLLRCS) in the Status register will be set. Detecting a stopped reference clock can take a long time, on the order of 2¹⁷ CLK_DFLL48M cycles. When the reference clock is stopped, the DFLL48M will operate as if in open-loop mode. Closed-loop mode operation will automatically resume if the CLK_DFLL48M_REF is restarted. An interrupt is generated on a zero-to-one transition on STATUS.DFLLRCS if the DFLL Reference Clock Stopped bit (INTENSET.DFLLRCS) in the Interrupt Enable Set register is set.

DFLL48M Low Frequency, Low-Power mode

The DFLL48M oscillator can operate in both open loop or closed loop at a reduced frequency of 8 MHz. To select the low frequency or low power set the Low Frequency bit (LOWFREQ) in the DFLL48M register. This bit is enable protected and can be changed only when the DFLL48M is

disabled. In closed-loop the user should adjust the multiplier value DFLLMUL.MUL depending on the frequency of the reference clock (CLK_DFLL48M_REF) in order to have an output clock frequency of 8 MHz.

19.5.4.2 Additional DFLL48M Features

Dealing with Delay in the DFLL48M in Closed-Loop Mode

Note: During a maximum 30 cycles of the reference clock period, between lock flag asserted and frequency stabilization, DFLL accuracy will be limited. To minimize the cycle-to-cycle jitter during lock search, in addition to selecting a low step value (DFLLMUL.FSTEP, DFLLMUL.CSTEP), it is recommended to disable the Quick Lock feature.

The time from selecting a new CLK_DFLL48M frequency until this frequency is output by the DFLL48M can be up to several microseconds. If the value in DFLLMUL.MUL is small, this can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving lock. To avoid this, a chill cycle, during which the CLK_DFLL48M frequency is not measured, can be enabled. The chill cycle is enabled by default, but can be disabled by writing a one to the DFLL Chill Cycle Disable bit (DFLLCTRLB.CCDIS) in the DFLL Control register. Enabling chill cycles might double the lock time.

Another solution to this problem consists of using less strict lock requirements. This is called Quick Lock (QL), which is also enabled by default, but it can be disabled by writing a one to the Quick Lock Disable bit (DFLLCTRLB.QLDIS) in the DFLL Control register. The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

DFLL48M Lose Lock After Wake

DFLL48M can optionally reset its lock bit when it is disabled or stopped (not requested). This is configured by the Lose Lock After Wake bit (DFLLCTRLB.LLAW) in the DFLL Control register. If DFLLCTRLB.LLAW is zero, when the DFLL48M is re-enabled or requested again, it starts running with the same configuration as before being disabled, even if the reference clock is not available. The lock will not be lost. Therefore it is important that the user checks that the DFLL48M has reached the lock stage before entering a sleep mode. When the reference clock has restarted, the TUNE tracking will quickly compensate for any frequency drift during sleep if DFLLCTRLB.STABLE is zero. If DFLLCTRLB.LLAW is one when disabling or stopping the DFLL48M, the DFLL48M will lose its lock and needs to regain it through the full lock sequence.

DFLL48M Wait for Lock

DFLL48M can optionally control the issued clock. This is configured by the Wait For Lock bit (DFLLCTRLB.WAITLOCK) in the DFLL Control register. If DFLLCTRLB.WAITLOCK is zero, the DFLL48M will issue a clock immediately after the ready bit (STATUS.DFLLRDY) has risen. If DFLLCTRLB.WAITLOCK is one, the DFLL48M will issue a clock immediately after the lock bit (STATUS.DFLLCK) has risen. Using the wait for lock feature allows a better accuracy of the issued DFLL48M clock, conversely it increases the startup time of the DFLL48M clock.

DFLL48M Accuracy

The following two main factors that determine the accuracy of DFLL48M which can be tuned to obtain maximum accuracy when fine lock is achieved.

- Resolution: The frequency step between two tune values.
- The accuracy of the reference clock.

DFLL48M Backup oscillator

The DFLL48M has an alternate 6 MHz backup oscillator. At any DFLL48M oscillator power-up, the 6 MHz backup oscillator is started and watches for DFLL48M oscillations. If no oscillation is present after the longest DFLL48M startup time, the DFLL48M clock is switched to the 6 MHz backup oscillator.

The DFLL Startup Failure bit in the STATUS register (STATUS.DFLLFAIL) will be set. The INTFLAG.DFLLFAIL bit is set on a zero-to-one transition of STATUS.DFLLFAIL and an interrupt is generated if the DFLL Startup Failure bit in the Interrupt Enable Set register (INTENSET.DFLLFAIL) is set.

19.5.5 Phase Locked Loop (PLL) Operation

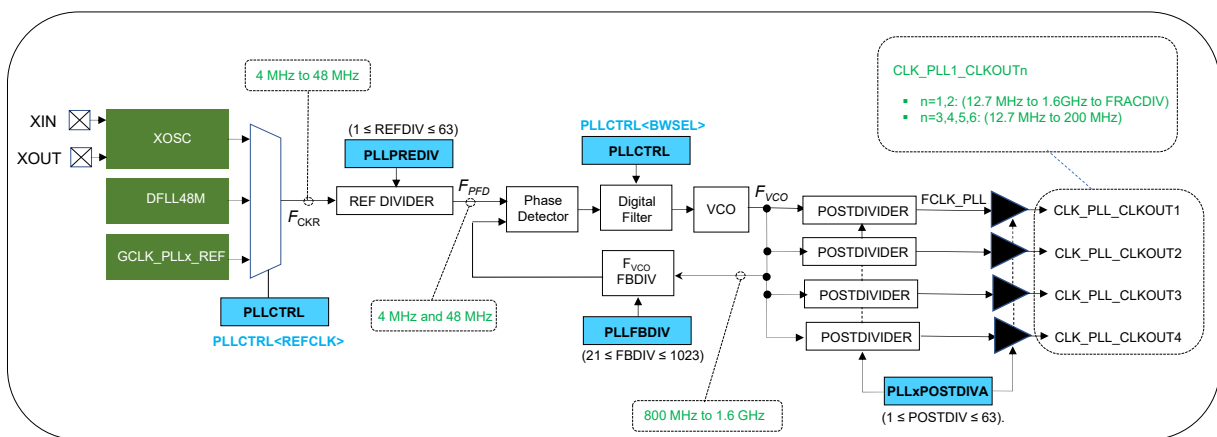
➔ Important: Before attempting to initialize and enable either of the two PLL's, the user must first enable the internal PLL regulator, SUPC.VREGCTRL.AVREGEN, and then wait a minimum of 55µs for internal power to stabilize before reading or writing any of the PLL registers. The internal BOOT ROM code may already have enabled the PLL so user should first check if PLL regulator has already been enabled.

- ➔ Important:** To achieve stable operation of the PLL as a primary clock source across temperature, the following procedure should be followed.
1. PLL0 must be dedicated to the CPU.
 2. PLL0 must be stepped up to the operating frequency in ≤ 75 MHz increments.
 3. The step delay needs to be ≥ 1 us.

The PLL provides a wide range of outputs from 12.7 MHz to 1600 MHz with, (i.e. FCLK_PLL max = 300MHz), support for input reference clock ranges from 4 MHz to 48 MHz. The PLL maintains a locked phase between the VCO input (reference) signal FPDF and the respective VCO output frequency FVCO via phase comparison and frequency multiplication.

The clocks from the PLL output(s), (CLK_PLLx_300n) is a source for the *Generic Clock module (GCLK)*.

Figure 19-2. PLL Block Diagram



➔ Important: The frequency generated by the PLL oscillator can be up to 1600 MHz but is limited to 300 MHz (max) for everything except inputs to FRACTIONAL DIVIDER. Depending on the operating conditions of the product using the PLL, the maximum allowed frequency can be as low as 12.7 MHz. Refer to the electrical characteristics of the product for a safe configuration of the PLL controller.

19.5.5.1 PLL Basic Operation

PLL Initialization, Enabling, and Disabling

The PLL is enabled by writing a one to the Enable bit in the Control register (PLLCTRL.ENABLE). The PLL is disabled by writing a zero to PLLCTRL.ENABLE.

➔ Important: If the PLL is active, the user must ensure that at no time do they violate the minimum or maximum frequency ranges specified for F_{PFD} , F_{VCO} , and F_{CLK_PLL} . Failure to do destabilize the VCO and lead to unpredictable behavior. Therefore carefully select the order in which the user configures the various scaler values in the PLL based on the FCKR clock source frequencies used in the application.

PLL Reference selection

The PLL controller supports three independent sources of reference clock:

- **DFLL48M:** The reference clock is provided by the DFLL48M.
- **XOSC:** The reference clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- **GCLK_PLLn_REF:** The reference clock "GCLK_PLLn_REF" is provided by the Generic Clock Controller.

The reference source can be selected by setting the REFSEL bits in the PLLCTRL register. The frequency of the reference must be comprised between 4 MHz and 48 MHz.

PLL settings

The frequency generated by the PLL is determined by the following control registers:

- **PLLREFDIV:** The PLL reference frequency divider, $1 \leq \text{REFDIV} \leq 63$
- **PLLFBDIV:** The PLL Feed Back frequency divider, $21 \leq \text{FBDIV} \leq 1023$
- **POSTDIV:** The PLL output frequency divider, $1 \leq \text{POSTDIV} \leq 63$

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is given in the equation below:

Equation 19-2. F_{CLK_PLL}

$$F_{CLK\_PLL} = (F_{CKR} * (FBDIV / (REFDIV * POSTDIV)))$$

Where,

F_{CLK_PLL} is the frequency of the PLL output clock, F_{CKR} is the frequency of the selected reference clock, REFDIV is the reference prescaler value, FBDIV is the loop divider value, and POSTDIV is the output prescaler value.

Note: F_{CLK_PLL} must always remain between 12.7 MHz to 300 MHz while configuring the various stages of the PLL, hence the VCO does not become unstable and lead to unpredictable behavior.

Example:

The goal is to have $F_{CLK\_PLL} = 300 \text{ MHz}$:

Given:

- $XOSC = 12 \text{ MHz} = F_{CKR}$
- If $REFDIV = 2$ then $F_{PDF} = (F_{CKR} / 2) = 6 \text{ MHz}$
- If $FBDIV = 150$ then $F_{VCO} = (F_{PDF} * FBDIV) = (6 \text{ MHz} * 150) = 900 \text{ MHz}$
- Setting $POSTDIV = 3$ then gives $F_{CLK\_PLL} = (F_{VCO} / 3) = (900 \text{ MHz} / 3) = 300 \text{ MHz}$

The frequency after the reference divider (FPDF) is given by the formula:

- $F_{PDF} = F_{CKR} / REFDIV$ (must be between 4 MHz to 48 MHz)

The frequency of the Voltage Controlled Oscillator (VCO) giving the PLL oscillation is given by the formula:

- $F_{VCO} = F_{CKR} * (FBDIV / REFDIV)$ (must be between 800 MHz and 1600 MHz)

Note: F_{CKR} , $REFDIV$ and $FBDIV$ must be selected to satisfy these conditions.

Notes:

1. The PLLREFDIV and PLLFBDIV registers are not write-protected by the PAC. Alternatively, they can be write protected by setting the control bit (PLLCTRL.WRTLOCK). When the OSCCTRL is PAC write-protected, the user can still tune the PLL frequency when the PLLCTRL.WRTLOCK is cleared, or the user can also write-protect the PLLREFDIV and PLLFBDIV registers when the PLLCTRL.WRTLOCK is set.
2. Each PLL has up to four outputs. Each output has an individual PLL output frequency divider POSTDIVn and an individual output control enable OUTENn with n=0..3. Depending on the selected REFDIV and the reference frequency, the user must set the Band Width selection bits (BWSEL) in the PLLCTRL register. Refer to the PLLCTRL.BWSEL definition in the "Register Description" section.

PLL Lock and Clock Generation

After the PLL oscillator is enabled, the PLL controller waits for the oscillator to issue a Lock status. The PLLLOCK bit in the status register (STATUS.PLLLOCK) will be set and the PLL controller issue the clock to the system (GCLK). The frequency of the PLL output clock CLK_PLL is stable when the STATUS.PLLLOCK bit is set. The PLL lock rising bit (PLLLOCKR) in the INTFLAG register is set when the STATUS.PLLLOCK rises.

Note: During each PLL start-up phase, the clock to the internal modules is not delivered as long as the first lock is not detected. When the lock is detected, the clock is released to the GCLK as long as requested. The PLL outputs whose OUTEN bit are set will start issuing a clock when the PLL LOCK status bit STATUS.PLLLOCK = 1 is set.

PLL Disabling

The PLL is disabled by writing a zero to PLLCTRL.ENABLE. Due to the synchronization of control and configuration registers, generation of internal timings to stop and power down properly the PLL, the PLL will be active for a few microseconds after CTRLA.ENABLE is cleared. The end of the PLL activity can be checked with the status lock bit STATUS.LOCK being ZERO. The PLL reference must not be stopped until the status lock bit STATUS.LOCK is read ZERO.

PLL Operation in Sleep Modes

The PLL will behave differently in different sleep modes, based on the settings of PLLCTRL.ONDEMAND and PLLCTRL.ENABLE. PLLCTRL.ONDEMAND must be written when PLLCTRL.ENABLE = 0. Otherwise, the write of this bit is ignored. If PLLCTRL.ENABLE = 0, the PLL will always be stopped. For PLLCTRL.ENABLE = 1, this table is valid:

Table 19-3. PLL Sleep Behavior

CPU Mode	ON DEMAND	Sleep Behavior of DFLL48M
Active or Idle	0	Always run
Active or Idle	1	Run if requested by a peripheral
Standby	0	Always run
Standby	1	Run if requested by a peripheral
Backup	0	Always OFF
Backup	1	Always OFF

PLL Reference Clock Switching

When a software operation requires reference clock switching, the normal operation is to disable the PLL, modify the PLLCTRL.REFCLK and PLLCTRL.BWSEL to select the desired reference source and activate the PLL again. The CLK_PLL output clock is ready when STATUS.PLLLOCK bit is set.

PLL Updates

It is important to note that when doing run time PLL updates that the user ALWAYS insures that for any PLL register update, the PLL does not exceed any of the electrical specs listed below to ensure continued stable PLL operation. Depending on the PLL input clock frequency selected therefore the user may have to choose a specific sequence, (i.e., order), of PLL register updates to maintain specification compliance throughout the PLL module logic.

- $F_{CKR} = 4 \text{ MHz to } 48 \text{ MHz}$
- $F_{PFD} = 4 \text{ MHz and } 48 \text{ MHz}$
- $F_{VCO} = 800 \text{ MHz to } 1.6 \text{ GHz}$
- $F_{CLK\_PLL} = 12.7 \text{ MHz to } 300 \text{ MHz}$

19.5.5.2 Fractional Divider

The Fractional Frequency Divider divides the PLL0 VCO, FVCO0, clock output frequency by a ratio composed of an integer part and a reminder part. Only the PLL0 supplies the clocks to the fractional dividers. FRACDIV0 is fed by PLL0 output 0 (under control of PLL0POSTDIVA[0]) and FRACDIV1 is fed by PLL0 output 1 (under control of PLL0POSTDIVA[1]). The maximum fractional divider input frequency is 1.6 GHz. The divided frequency is given by the integer and reminder part of the divider, FRACDIV.INTDIV and FRACDIV.REMDIV. The resulting frequency FFRACDIV is calculated using the following equation:

Equation 19-3. Fractional Divider Frequency

$$F_{CLK\_PLL0\_FRC\_CLKOUTn} = (F_{CLK\_PLL} / (2 * (INTDIV + (REMDIV / 512))))$$

Notes: This is not a true fractional divider in the sense that the resulting frequency is actually an average over time that represents the fractional frequency except in the cases where $F_{CLK\_PLL0\_FRC\_CLKOUTn}$ value corresponds to a whole integer value. For decimal fractional values, the fractional divider logic steals input F_{VCO} PLL clock cycles to produce and average output frequency equivalent to the desired frequency. As a result, the final $F_{CLK\_PLL0\_FRC\_CLKOUTn}$ output frequency will have jitter equivalent to:

- If $\text{Reminder} \leq 0.5$:
 - $F_{CLK\_PLL0\_FRC\_CLKOUTn} \text{ Jitter} = (\text{Remainder} / F_{VCO})$
- If $\text{Reminder} > 0.5$:
 - $F_{CLK\_PLL0\_FRC\_CLKOUTn} \text{ Jitter} = ((1 - \text{Remainder}) / F_{VCO})$

Fractional Divider Operating mode

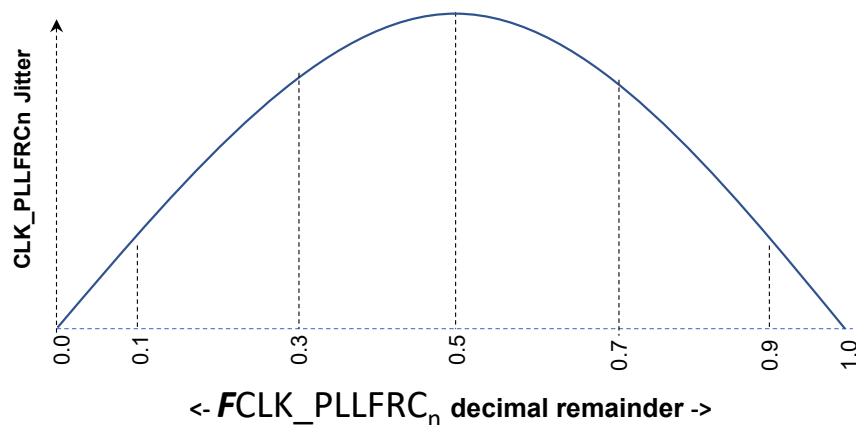
The Fractional Divider must be configured and enabled as shown in the following section.

19.5.5.2.1 Configure and Enable the PLL Output

1. Configure the integer and reminder divider factors FRACDIV.INTDIV and FRACDIV.REMDIV. If the PLL is not locked, the FRACDIV register write is pending until the PLL deliver a clock after the lock. The pending status can be checked in the SYNCBUSY.FRACDIVn register bit. The FRACDIV register can still be modified if the source PLL enable bit is not set.
2. Enable the PLL output.
3. Check the PLL lock with the register bit INTFLAG.PLLnLOCKR.
4. Check the SYNCBUSY.FRACDIVn register bit. When this bit is low, the FRACDIV is delivering the divided PLL clock. The Fractional Divider values FRACDIV.INTDIV and FRACDIV.REMDIV can be changed when the divider is operating. But before, the user must ensure that the previous Fractional Divider values change is completed by checking the bit SYNCBUSY.FRACDIVn is low.

Note: After a system reset the Fractional Divider starts in a frozen state. If the Fractional Divider is operating or synchronizing the divider factors, it is requesting a clock to the PLL source. The FRACDIV will stay frozen until the PLL source delivers a clock. After the PLL source delivers a clock, the FRACDIV is unfrozen. If the fractional Divider stops its operation (no more GCLK request and no more on-going synchronization), it will go back to a frozen state and request the PLL source clock until it is actually frozen. The freeze/unfreeze process ensures the FRACDIV divides on a locked PLL clock.

Figure 19-3. Fractional Divider



Notes:

1. $FCLK\_PLLFR0 = (F_{CLK\_PLL0} / (2 \times (INTDIV + (REMDIV / 512))))$.
2. The maximum permitted fractional output frequency, FCLK_PLLFR0, must always be limited to 200MHz by the user.
3. Setting both INTDIV and REMDIV = 0 will yield $FCLK\_PLLFR0 = F_{CLK\_PLL0}$ which effectively bypasses the fractional divider module in which case the user must limit the output of PLL0 to the fractional divider module to $F_{PLL0} = F_{CLK\_PLL0} = 200MHz$ max.

19.5.6 OSCCTRL Interrupts

The OSCCTRL has the following interrupt sources:

- **XOSCRDY - Multipurpose Crystal Oscillator Ready:** A "0-to-1" transition on the STATUS.XOSCRDY bit is detected
- **XOSCFAIL - Xosc Startup Failure:** A "0-to-1" transition on the STATUS.XOSCFAIL bit is detected
- **CLKFAIL - Xosc Clock Failure:** A "0-to-1" transition on the STATUS.CLKFAIL bit is detected
- **DFLLRDY - DFLL48m Ready:** A "0-to-1" transition on the STATUS.DFLLRDY bit is detected

- **DFLLLOCK - DFLL48m Lock:** A "0-to-1" transition on the STATUS.DFLLLOCK bit is detected
- **DFLLOVF - DFLL48m Overflow:** A "0-to-1" transition on the STATUS.DFLOVF bit is detected
- **DFLLUNF - DFLL48m Underflow:** A "0-to-1" transition on the STATUS.DFLLUNF bit is detected
- **DFLLRCS - DFLL48m Reference Clock Stop:** A "0-to-1" transition on the STATUS.DFLLRCS bit is detected
- **DFLLFAIL - DFLL Startup Failure:** A "0-to-1" transition on the STATUS.DFLLFAIL bit is detected

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set, and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has fewer request lines than interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

19.5.7 OSCCTRL Events

The Clock Fail Detect (CFD) can generate the following output event:

- **Clock Failure (CLKFAIL):** Generated when the XOSC Clock Failure status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the XOSC Clock Switch bit (STATUS.XOSCCKSW) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

19.5.8 OSCCTRL Synchronization

Due to the multiple clock domains, some registers in the DFLL48M must be synchronized when accessed. A register can require:

- Synchronization when written
- Synchronization when read
- No synchronization when executing an operation that requires synchronization, the relevant Synchronization bit in the Synchronization Busy register (DFLLSYNC) will be set immediately and cleared when synchronization is complete

The following registers need synchronization:

- The ENABLE bit in the DFLLCTRLA register - write-synchronized
- The DFLLCTRLB register - read-synchronized
- The DFLLTUNE register - read-synchronized and write-synchronized
- The DFLLMUL register - write-synchronized

19.6 Peripheral Dependencies

Table 19-4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
OSCCTRL	0x44006000	4 : XOSCRDY/ XOSCFAIL/ CLKFAIL 5 : DFLLRDY/ DFLLLOCK/ DFLLOVF/ DFLLUNF/ DFLLRCS/ DFLLFAIL 6 : PLLLOCKR_0/ PLLLOCKF_0	APB: CLKMSK[0] MASK6	DFLL48M: PCHCTRL[0] PLL: PCHCTRL[1]	6 STATUS0 PERID6		<u>Generators:</u> XOSCFAIL: CHANNEL[2]	PD_CORE_SW

19.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								CFDEO
0x04	INTENCLR	31:24							PLLxLOCKF	PLLxLOCKR
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0						CLKFAIL	XOSCFAIL	XOSCRDY
0x08	INTENSET	31:24							PLLxLOCKF	PLLxLOCKR
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0						CLKFAIL	XOSCFAIL	XOSCRDY
0x0C	INTFLAG	31:24							PLLxLOCKF	PLLxLOCKR
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0						CLKFAIL	XOSCFAIL	XOSCRDY
0x10	STATUS	31:24								PLLxLOCK
		23:16								
		15:8			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
		7:0					XOSCKSW	CLKFAIL	XOSCFAIL	XOSCRDY
0x14	XOSCCTRLA	31:24	WRTLOCK						USBHSDIV[1:0]	
		23:16						CFDPPRESC[3:0]		
		15:8						STARTUP[3:0]		
		7:0	ONDEMAND		SWBEN	CFDEN	XTALEN	AGC	ENABLE	
0x18	XOSCCTRLB	31:24	WRTLOCK							
		23:16								
		15:8								
		7:0								USRCFG[7:0]
0x1C ... 0x2B	Reserved									
0x2C	DFLLCTRLA	31:24								
		23:16								
		15:8								
		7:0	ONDEMAND				LOWFREQ	WRTLOCK	ENABLE	
0x30	DFLLCTRLB	31:24								
		23:16								
		15:8								
		7:0	WAITLOCK		QLDIS	CCDIS	USBCRM	LLAW	STABLE	LOOPEN
0x34	DFLLTUNE	31:24								
		23:16								
		15:8								
		7:0								TUNE[6:0]
0x38	DFLLDIFF	31:24								
		23:16								
		15:8								DIFF[15:8]
		7:0								DIFF[7:0]
0x3C	DFLLMUL	31:24								
		23:16								STEP[6:0]
		15:8								MUL[15:8]
		7:0								MUL[7:0]
0x40	PLLOCTRL	31:24								
		23:16								
		15:8					BWSEL[2:0]		REFSEL[2:0]	
		7:0	ONDEMAND					WRTLOCK	ENABLE	
0x44	PLLOFBDIV	31:24								
		23:16								
		15:8								FBDIV[9:8]
		7:0								FBDIV[7:0]

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x48	PLL0REFDIV	31:24								
		23:16								
		15:8								
		7:0			REFDIV[5:0]					
0x4C	PLL0POSTDIVA	31:24								
		23:16								
		15:8								
		7:0	OUTEN		POSTDIV[5:0]					
0x50	PLL0POSTDIVB	31:24								
		23:16								
		15:8								
		7:0	OUTEN		POSTDIV[5:0]					
0x54	PLL1CTRL	31:24								
		23:16								
		15:8			BWSEL[2:0]			REFSEL[2:0]		
		7:0	ONDEMAND					WRTLOCK	ENABLE	
0x58	PLL1FBDIV	31:24								
		23:16								
		15:8							FBDIV[9:8]	
		7:0	FBDIV[7:0]							
0x5C	PLL1REFDIV	31:24								
		23:16								
		15:8								
		7:0			REFDIV[5:0]					
0x60	PLL1POSTDIVA	31:24								
		23:16								
		15:8								
		7:0	OUTEN		POSTDIV[5:0]					
0x64	PLL1POSTDIVB	31:24								
		23:16								
		15:8								
		7:0	OUTEN		POSTDIV[5:0]					
0x68 ... 0x6B	Reserved									
0x6C	FRACDIV0	31:24		INTDIV[14:8]						
		23:16		INTDIV[7:0]						
		15:8		REMDIV[8:1]						
		7:0	REMDIV[0]							
0x70 ... 0x73	Reserved									
0x74	FRACDIV1	31:24		INTDIV[14:8]						
		23:16		INTDIV[7:0]						
		15:8		REMDIV[8:1]						
		7:0	REMDIV[0]							
0x78	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0	FRACDIV1	FRACDIV0	DFLLMUL	DFLLDIFF	DFLLTUNE	DFLLCTRLB	DFLLENABLE	
0x7C ... 0x7F	Reserved									
0x80	XOSCCAL	31:24								
		23:16								
		15:8	CAL[15:8]							
		7:0	CAL[7:0]							
0x84 ... 0x87	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x88	RC48MCAL0	31:24	CAL[31:24]							
		23:16	CAL[23:16]							
		15:8	CAL[15:8]							
		7:0	CAL[7:0]							

19.7.1 Event Control

Name: EVCTRL
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 19-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								CFDEO
Access								RW
Reset								0

Bit 0 – CFDEO Clock Failure Detector Event Output Enable

This bit indicates whether the XOSC Clock Failure detector event output is enabled and an output event will be generated when the XOSC Clock Failure detector detects a clock failure.

Note: To prevent false event generation, the bit CFDEO must be set or cleared only when the XOSC is disabled (XOSCCTRLn.ENABLE=0).

Value	Description
0	Clock Failure detector event output is disabled and an event will not be generated on a clock fail.
1	Clock Failure detector event output is enabled and an event will be generated on a clock fail.

19.7.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 19-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							PLLxLOCKF	PLLxLOCKR
Access							RW	RW
Reset							0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						CLKFAIL	XOSCFail	XOSCRDY
Access						RW	RW	RW
Reset						0	0	0

Bit 25 – PLLxLOCKF PLL x Lock Fall Interrupt Enable

Bit 24 – PLLxLOCKR PLL x Lock Rise Interrupt Enable

Bit 13 – DFLLFAIL DFLL Startup Failure Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Startup Failure Interrupt Enable bit, which disables the DFLL Startup Failure interrupt.

Value	Description
0	The DFLL48M Startup Failure interrupt is disabled.
1	The DFLL48M Startup Failure interrupt is enabled, and an interrupt request will be generated when the DFLL Startup Failure Interrupt flag is set.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL48M Reference Clock Stopped interrupt is disabled.

Value	Description
1	The DFLL48M Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLUNF DFLL Tuner Underflow Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Tuner Underflow Interrupt Enable bit, which disables the DFLL Tuner Underflow interrupt.

Value	Description
0	The DFLL Tuner Underflow interrupt is disabled.
1	The DFLL Tuner Underflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Underflow Interrupt flag is set.

Bit 10 – DFLLLOVF DFLL Tuner Overflow Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Tuner Overflow Interrupt Enable bit, which disables the DFLL Tuner Overflow interrupt.

Value	Description
0	The DFLL Tuner Overflow interrupt is disabled.
1	The DFLL Tuner Overflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Overflow Interrupt flag is set.

Bit 9 – DFLLLOCK DFLL Lock Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Lock Interrupt Enable bit, which disables the DFLL Lock interrupt.

Value	Description
0	The DFLL Lock interrupt is disabled.
1	The DFLL Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set.

Bit 8 – DFLLRDY DFLL Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 2 – CLKFAIL XOSC Clock Failure Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

Value	Description
0	The XOSC Clock Failure interrupt is disabled.
1	The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 1 – XOSCFAIL XOSC Startup Failure Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC Startup Failure Interrupt Enable bit, which disables the XOSC Startup Failure interrupt.

Value	Description
0	The XOSC Startup Failure interrupt is disabled.
1	The XOSC Startup Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Startup Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

19.7.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x08
Reset: 0x00000000
Property: RW

Table 19-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							PLLxLOCKF	PLLxLOCKR
Access							RW	RW
Reset							0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						CLKFAIL	XOSCFail	XOSCRDY
Access						RW	RW	RW
Reset						0	0	0

Bit 25 – PLLxLOCKF PLL x Lock Fall Interrupt Enable

Bit 24 – PLLxLOCKR PLL x Lock Rise Interrupt Enable

Bit 13 – DFLLFAIL DFLL Startup Failure Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Startup Failure Interrupt Enable bit, which disables the DFLL Startup Failure interrupt.

Value	Description
0	The DFLL48M Startup Failure interrupt is disabled.
1	The DFLL48M Startup Failure interrupt is enabled, and an interrupt request will be generated when the DFLL Startup Failure Interrupt flag is set.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

Value	Description
0	The DFLL48M Reference Clock Stopped interrupt is disabled.

Value	Description
1	The DFLL48M Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.

Bit 11 – DFLLUNF DFLL Tuner Underflow Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Tuner Underflow Interrupt Enable bit, which disables the DFLL Tuner Underflow interrupt.

Value	Description
0	The DFLL Tuner Underflow interrupt is disabled.
1	The DFLL Tuner Underflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Underflow Interrupt flag is set.

Bit 10 – DFLLLOVF DFLL Tuner Overflow Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Tuner Overflow Interrupt Enable bit, which disables the DFLL Tuner Overflow interrupt.

Value	Description
0	The DFLL Tuner Overflow interrupt is disabled.
1	The DFLL Tuner Overflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Overflow Interrupt flag is set.

Bit 9 – DFLLLOCK DFLL Lock Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Lock Interrupt Enable bit, which disables the DFLL Lock interrupt.

Value	Description
0	The DFLL Lock interrupt is disabled.
1	The DFLL Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set.

Bit 8 – DFLLRDY DFLL Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

Value	Description
0	The DFLL Ready interrupt is disabled.
1	The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.

Bit 2 – CLKFAIL XOSC Clock Failure Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

Value	Description
0	The XOSC Clock Failure interrupt is disabled.
1	The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set.

Bit 1 – XOSCFAIL XOSC Startup Failure Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC Startup Failure Interrupt Enable bit, which disables the XOSC Startup Failure interrupt.

Value	Description
0	The XOSC Startup Failure interrupt is disabled.
1	The XOSC Startup Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Startup Failure Interrupt flag is set.

Bit 0 – XOSCRDY XOSC Ready Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

Value	Description
0	The XOSC Ready interrupt is disabled.
1	The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set.

19.7.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x00000000
Property: RW

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 19-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							PLLxLOCKF	PLLxLOCKR
Access							RW	RW
Reset							0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						CLKFAIL	XOSCFail	XOSCRDY
Access						RW	RW	RW
Reset						0	0	0

Bit 25 – PLLxLOCKF PLL x Lock Fall

Bit 24 – PLLxLOCKR PLL x Lock Rise

Bit 13 – DFLLFAIL DFLL Startup Failure

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the DFLL Startup Failure bit in the Status register (STATUS.DFLLFAIL) and will generate an interrupt request if INTENSET.DFLLFAIL is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DFLL Startup Failure interrupt flag.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS.DFLLRCS) and will generate an interrupt request if INTENSET.DFLLRCS is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

Bit 11 – DFLLUNF DFLL Tuner Underflow

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the DFLL Tuner Underflow bit in the Status register (STATUS.DFLLUNF) and will generate an interrupt request if INTENSET.DFLLUNF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DFLL Tuner Underflow interrupt flag.

Bit 10 – DFLOVF DFLL Tuner Overflow

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the DFLL Tuner Overflow bit in the Status register (STATUS.DFLOVF) and will generate an interrupt request if INTENSET.DFLOVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DFLL Tuner Overflow interrupt flag.

Bit 9 – DFLLLOCK DFLL Lock

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the DFLL Lock bit in the Status register (STATUS.DFLLLOCK) and will generate an interrupt request if INTENSET.DFLLLOCK is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DFLL Lock interrupt flag.

Bit 8 – DFLLRDY DFLL Ready

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the DFLL Ready interrupt flag.

Bit 2 – CLKFAIL XOSC Clock Failure

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the XOSC Clock Failure bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the XOSC Clock Failure interrupt flag.

Bit 1 – XOSCFAIL XOSC Startup Failure

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the XOSC Startup Failure bit in the Status register (STATUS.XOSCFAIL) and will generate an interrupt request if INTENSET.XOSCFAIL is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the XOSC Startup Failure interrupt flag.

Bit 0 – XOSCRDY XOSC Ready

This flag is cleared by writing a '1' to it.

This flag is set on a zero-to-one transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the XOSC Ready interrupt flag.

19.7.5 Status

Name: STATUS
Offset: 0x10
Reset: 0x00000000
Property: R

Table 19-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PLLxLOCK
Access								R
Reset								0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			DFLLFAIL	DFLLRCS	DFLLUNF	DFLLOVF	DFLLLOCK	DFLLRDY
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					XOSCCKSW	CLKFAIL	XOSCFail	XOSCRDY
Access					R	R	R	R
Reset					0	0	0	0

Bit 24 – PLLxLOCK PLL x Lock

Bit 13 – DFLLFAIL DFLL Startup Failure

Value	Description
0	DFLL Startup failure is not detected.
1	DFLL Startup failure is detected.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped

Value	Description
0	DFLL reference clock is running.
1	DFLL reference clock has stopped.

Bit 11 – DFLLUNF DFLL Tuner Underflow

Value	Description
0	DFLL Tuner Underflow is not detected.
1	DFLL Tuner Underflow is detected.

Bit 10 – DFLLOVF DFLL Tuner Overflow

Value	Description
0	DFLL Tuner Overflow is not detected.
1	DFLL Tuner Overflow is detected.

Bit 9 – DFLLLOCK DFLL Lock

Value	Description
0	DFLL lock is not detected.
1	DFLL lock is detected.

Bit 8 – DFLLRDY DFLL Ready

Value	Description
0	DFLL is not ready.
1	DFLL is stable and ready to be used as a clock source.

Bit 3 – XOSCCKSW XOSC Clock Switch

Value	Description
0	XOSC is not switched and provides the external clock or crystal oscillator clock.
1	XOSC is switched and provides the safe clock.

Bit 2 – CLKFAIL XOSC Clock Failure

Value	Description
0	XOSC Clock failure is not detected.
1	XOSC Clock failure is detected.

Bit 1 – XOSCFAIL XOSC Startup Failure

Value	Description
0	XOSC Startup failure is not detected.
1	XOSC Startup failure is detected.

Bit 0 – XOSCRDY XOSC Ready

Note: If the CFD is enabled and the XOSC clock is failing, the XOSCRDY status bit remains high, if already set.

Value	Description
0	XOSC is not ready.
1	XOSC is stable and ready to be used as a clock source.

19.7.6 External Multipurpose Crystal Oscillator Control A

Name: XOSCCTRLA
Offset: 0x14
Reset: 0x00000D00
Property: RW

Table 19-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WRTLOCK						USBHSDIV[1:0]	
Access	RW						RW	RW
Reset	0						0	0
Bit	23	22	21	20	19	18	17	16
						CFDPRESC[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
						STARTUP[3:0]		
Access					RW	RW	RW	RW
Reset					1	1	0	1
Bit	7	6	5	4	3	2	1	0
	ONDEMAND		SWBEN	CFDEN	XTALEN	AGC	ENABLE	
Access	RW		RW	RW	RW	RW	RW	
Reset	0		0	0	0	0	0	

Bit 31 – WRTLOCK Write Lock for CTRLA register

Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The XOSCCTRLA register can be modified by a system write.
1	The XOSCCTRLA (except XOSCCTRLA.SWBEN) register is write protected.

Bits 25:24 – USBHSDIV[1:0] USBHS Reference Clock Division

Value	Name	Description
0	DIS	USBHS PLL reference XOSC clock is disabled
1	DIV1	USBHS PLL reference XOSC clock is divided by 1
2	DIV2	USBHS PLL reference XOSC clock is divided by 2
3	DIV4	USBHS PLL reference XOSC clock is divided by 4

Bits 19:16 – CFDPRESC[3:0] Clock Failure Detector Prescaler

These bits select the DFLL48oscillator post scaler for the clock fail detector. The CFD safe clock frequency is the DFLL48 frequency divided by $2^{CFDPRESC}$. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE=1

Value	Name	Description
0	DIV1	48 MHz

Value	Name	Description
1	DIV2	24 MHz
2	DIV4	12 MHz
3	DIV8	6 MHz
4	DIV16	3 MHz
5	DIV32	1.5 MHz
6	DIV64	0.75 MHz
7	DIV128	0.375 MHz
8	DIV256	187.5 KHz
9	DIV512	93.75 KHz
10	DIV1024	46.875 KHz
11	DIV2048	23.437 KHz
12	DIV4096	11.718 KHz
13	DIV8192	5.85 KHz
14	DIV16384	2.92 KHz
15	DIV32768	1.46 KHz

Bits 11:8 – STARTUP[3:0] Start-Up Time

These bits select start-up time for the oscillator XOSC according to the table below before a clock fail is acknowledged. The OSCULP32K oscillator is used to clock the start-up counter. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.

STARTUP[3:0]	Number of OSCULP32K Clock Cycles	Approximate Equivalent Time
0x0	1	31µs
0x1	2	61µs
0x2	4	122µs
0x3	8	244µs
0x4	16	488µs
0x5	32	977µs
0x6	64	1953µs
0x7	128	3906µs
0x8	256	7813µs
0x9	512	15625µs
0xA	1024	31250µs
0xB	2048	62500µs
0xC	4096	125000µs
0xD (Default)	8192	250000µs
0xE	16384	500000µs
0xF	32768	1000000µs

Notes:

1. It is critical when using AGC to allow ample startup time to avoid spurious CFD events.
2. When using AGC BW = 0x0, the minimum startup time is 6.25 ms.

Value	Name	Description
0	CYCLE1	31 us
1	CYCLE2	61 us
2	CYCLE4	122 us
3	CYCLE8	244 us
4	CYCLE16	488 us
5	CYCLE32	977 us
6	CYCLE64	1953 us
7	CYCLE128	3906 us
8	CYCLE256	7813 us
9	CYCLE512	15625 us

Value	Name	Description
10	CYCLE1024	31250 us
11	CYCLE2048	62500 us
12	CYCLE4096	125000 us
13	CYCLE8192	250000 us
14	CYCLE16384	500000 us
15	CYCLE32768	1000000 us

Bit 7 – ONDEMAND On Demand Control

The ONDEMAND operation mode allows the XOSC to be enabled or disabled depending on peripheral clock requests.

Note: The XOSC is not running if no peripheral is requesting the clock source.

If ONDEMAND is set, the XOSC will only be running when requested by a peripheral and enabled (XOSCCTRLA.ENABLE = 1). If there is no peripheral requesting the XOSC's clock source, the XOSC will be in a disabled state. If ONDEMAND is disabled, the XOSC will always be running when enabled (XOSCCTRLA.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active. This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.

Value	Description
0	The XOSC is always on.
1	The XOSC is running when a peripheral is requesting the XOSC to be used as a clock source.

Bit 5 – SWBEN Xosc Clock Switch Back Enable

This bit controls the XOSC output clock switch back to the external clock or crystal oscillator in case of clock recovery.

Note: The SWBEN bit is also cleared when the Clock Failure Detector is disabled (CFDEN = 0).

Value	Description
0	The clock switch back is disabled.
1	The clock switch back is enabled. This bit is reset once the XOSC output clock is switched back to the external clock or crystal oscillator.

Bit 4 – CFDEN Clock Failure Detector Enable

This bit controls the XOSC clock failure detector and is enable protected

Note: After setting CFDEN to enable clock failure detection, STATUS.CLKFAIL will always be set. This first detection must be ignored. Subsequent setting of this bit will indicate actual clock failure events.

Value	Description
0	Clock Failure Detector is disabled.
1	Clock Failure Detector is enabled.

Bit 3 – XTALEN Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator XOSC.

Notes:

1. If XOSCCTRLA.XTALEN = 0 then XOSCCTRLA.AGC = 1 is not permitted.
2. This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.
3. If XOSCCTRLA.XTALEN = 0 then XOSCCTRLB.GMAN (User Manual Gain control) bits are ignored.

Value	Description
0	External clock oscillator connected on XIN. XOUT can be used as general-purpose I/O.
1	Crystal connected to XIN and XOUT.

Bit 2 – AGC Auto Gain Control Loop Enable

Notes:

- If XOSCCTRLA.XTALEN = 0 then XOSCCTRLA.AGC = 1 is not permitted.
- This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.
- If AGC is enabled, XOSCCTRLB.GMAN (User Manual Gain control) bits are ignored.
- When the XOSCCTRLA.AGC = 1, the Primary Oscillator will automatically do a linear search to find the lowest power/gain setting to guarantee stable oscillation with the user's crystal.

Value	Description
0	The oscillator auto gain control loop is disabled.
1	The oscillator auto gain control loop is enabled.

Bit 1 – ENABLE Oscillator Enable

Value	Description
0	The oscillator XOSC is disabled.
1	The oscillator XOSC is enabled.

19.7.7 External Multipurpose Crystal Oscillator Control B

Name: XOSCCTRLB
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 19-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WRTLOCK							
Access								
Reset	0							
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	USRCFG[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – WRTLOCK Write Lock for CTRLB register

Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The XOSCCTRLB register can be modified by a system write.
1	The XOSCCTRLB register is write protected.

Bits 7:0 – USRCFG[7:0] User Configuration Control Bits

19.7.8 DFLL48M Control A

Name: DFLLCTRLA
Offset: 0x2C
Reset: 0x00000082
Property: RW

Table 19-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	ONDEMAND				LOWFREQ	WRTLOCK	ENABLE	
Access	RW				RW	RW	RW	
Reset	1				0	0	1	

Bit 7 – ONDEMAND On Demand Control

The ONDEMAND operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests.

Note: If ONDEMAND is set, the DFLL will only be running when requested by a peripheral and enabled (DFLLTRLA.ENABLE = 1). If there is no peripheral requesting the DFLL's clock source, the DFLL will be in a disabled state. If ONDEMAND is disabled the DFLL will always be running when enabled (DFLLTRLA.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active. This bit is DFLLCTRLA.ENABLE protected and cannot be updated if DFLLCTRLA.ENABLE = 1.

Value	Description
0	The DFLL is always on.
1	The DFLL is running when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is not running if no peripheral is requesting the clock source.

Bit 3 – LOWFREQ Low Frequency Mode

Note: This bit is DFLLCTRLA.ENABLE protected and cannot be updated if DFLLCTRLA.ENABLE = 1.

Value	Description
0	The DFLL48M oscillator operates at high frequency.
1	The DFLL48M oscillator operates at low frequency.

Bit 2 – WRTLOCK Write Lock

Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The DFLLCTRLA and DFLLCTRLB registers can be modified by a system write.
1	The DFLLCTRLA and DFLLCTRLB registers are write protected.

Bit 1 – ENABLE DFLL Enable

Note: This bit is write-synchronized: Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRLA.ENABLE will read back immediately after written.

Value	Description
0	The DFLL48M oscillator is disabled.
1	The DFLL48M oscillator is enabled.

19.7.9 DFLL48M Control B

Name: DFLLCTRLB
Offset: 0x30
Reset: 0x00000000
Property: RW

Note: In the period of 80-100 us between lock and stabilization, DFLL accuracy will be limited to +/-1.5%. After stabilization has been achieved, the accuracy will be +/-0.25%. Disabling Quick Lock will eliminate this period of inaccuracy.

Table 19-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	WAITLOCK		QLDIS	CCDIS	USBCRM	LLAW	STABLE	LOOPEN
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		0	0	0	0	0	0

Bit 7 – WAITLOCK Wait Lock

This bit controls the DFLL48M output clock, depending on lock status:

Value	Description
0	Output clock before the DFLL is locked.
1	Output clock when DFLL is locked.

Bit 5 – QLDIS Quick Lock Disable

Value	Description
0	Quick Lock is enabled.
1	Quick Lock is disabled.

Bit 4 – CCDIS Chill Cycle Disable

Value	Description
0	Chill Cycle is enabled.
1	Chill Cycle is disabled.

Bit 3 – USBCRM USB Clock Recovery Mode

Value	Description
0	USB Clock Recovery Mode is disabled.
1	USB Clock Recovery Mode is enabled.

Bit 2 – LLAW Lose Lock After Wake

Value	Description
0	Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1	Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

Bit 1 – STABLE Stable DFLL Frequency

Value	Description
0	Tune register tracks changes in output frequency.
1	Tune calibration register value will be fixed after a lock.

Bit 0 – LOOPEN Operating Mode Selection

Value	Description
0	The DFLL operates in open-loop operation.
1	The DFLL operates in closed-loop operation.

19.7.10 DFLL48M Tune

Name: DFLLTUNE
Offset: 0x34
Reset: 0x00000000
Property: RW

Table 19-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
		TUNE[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 6:0 – TUNE[6:0] Tune Value

Sets the value of the Tune Calibration register.

Note: In closed-loop mode, this field is read-only.

Step	TUNE[6:0]	% Delta/step
+63	0b011 1111	+9.45%
...	...	...
+1	0b000 0001	+0.15%
0	0b000 0000 / 0b111 1111	0%
-1	0b111 1110	-0.15%
...	...	...
-63	0b100 0000	-9.45%

Note:

1. % Delta value is rounded to two decimal places.

19.7.11 DFLL48M Diff

Name: DFLLDIFF
Offset: 0x38
Reset: 0x00000000
Property: R

Table 19-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	DIFF[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DIFF[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – DIFF[15:0] Multiplication Ratio Difference

In closed-loop mode (DFLLCTRLB.LOOPEN is written to one), this bit group indicates the difference between the ideal number of DFLL48M cycles and the counted number of cycles. This value is not updated in open loop mode and should be considered invalid in that case.

19.7.12 DFLL48M Multiplier

Name: DFLLMUL
Offset: 0x3C
Reset: 0x00000000
Property: RW

Table 19-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
		STEP[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MUL[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MUL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 22:16 – STEP[6:0] Tune Maximum Step

This bit group indicates the maximum step size allowed during tune adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

Note: STEP[6:0] is nothing more than the first tuner adjustment value from where the tune search will start. Depending if the current DFLL frequency is slower or faster than the targeted frequency, the next tune register value is *tune+step* or *tune-step*. Each time the current frequency changes from slower to faster or faster to slower, step is divided by 2 until the minimum value of 1.

Example 1:

If STEP[6:0] = 64, dichotomic search will be:

- step1 = +/-64
- step2 = +/-32
- step3 = +/-16
- step4 = +/-8
- step5 = +/-4
- step6 = +/-2
- step7 = +/-1

Example 2:

If STEP[6:0] = 15, dichotomic search will be:

- step1 = +/-15
- step2 = +/-7
- step3 = +/-3
- step4 = +/-1

Bits 15:0 – MUL[15:0] DFLL Multiply Factor

This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause the lock to be lost and the DFLLTUNE.TUNE register value to be reset to its midpoint, 0b000_0000.

Example: CLK_DFLL_REF = XTAL, 32.768 kHz, 100 ppm, DFLLMUL.MUL[15:0] = 1464 = 0x5B8, then DFLL48 = 47.97 MHz.

19.7.13 PLL Control

Name: PLL0CTRL
Offset: 0x40
Reset: 0x00000000
Property: RW

Table 19-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			BWSEL[2:0]			REFSEL[2:0]		
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ONDEMAND					WRTLOCK	ENABLE	
Access	RW					RW	RW	
Reset	0					0	0	

Bits 13:11 – BWSEL[2:0] Bandwidth selection

These bits select the PLL closed loop filter bandwidth, depending on the frequency after the reference divider F_{PFD} as shown in the table below. Selecting the correct filter bandwidth is important to operate the PLL/VCO in its best range.

Table 19-18. PLL0 BWSEL Filter

F_{PFD}	BWSEL[2:0]
Reserved	0b000
$4\text{MHz} \leq F_{PFD} < 10\text{MHz}$	0b001
$10\text{MHz} \leq F_{PFD} < 20\text{MHz}$	0b010
$20\text{MHz} \leq F_{PFD} < 30\text{MHz}$	0b011
$30\text{MHz} \leq F_{PFD} < 60\text{MHz}$	0b100
Reserved	0b101 – 0b111

Notes:

1. F_{PFD} is the frequency of the reference clock divided by the PLL0 reference divider PLLREFDIV.REFDIV. These bits are PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE = 1.
2. At elevated temperatures, the effective range of the Bandwidth setting will skew higher. Depending on the input frequency and operating temperature, it may be optimal to change the BWSEL setting to the next higher value.

Value	Name	Description
0x0	BWSEL0	TBD
0x1	BWSEL1	TBD
0x2	BWSEL2	TBD
0x3	BWSEL3	TBD
0x4	BWSEL4	TBD
0x5	BWSEL5	TBD
0x6	BWSEL6	TBD
0x7	BWSEL7	TBD

Bits 10:8 – REFSEL[2:0] Reference selection

These bits select the PLL0 clock reference, as shown in the table below.

REFSEL[2:0]	Selectedsource	Description
0x0 ^(2,3)	GCLK	Dedicated GCLK_PLL0_REF clock reference
0x1	XOSC	XOSC clock reference
0x2	DFLL48M	DFLL48M clock reference
0x3- 0x7	n/a	Reserved

Notes:

1. These bits are PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE = 1.
2. **IMPORTANT:** If GCLK source is PLL0 then you should not use GCLK_PLL0_REF as input clock source to PLL0. It would create a circular reference, an unstable clock and unexpected behavior.
3. The recommended clock sources for PLL0 are XOSC and the DFLL48M. Using the GCLK as a source is not recommended.

Value	Name	Description
0x0	GCLK	Dedicated GCLK clock reference
0x1	XOSC	XOSC clock reference
0x2	DFLL48M	DFLL48M clock reference

Bit 7 – ONDEMAND On Demand Control

The ONDEMAND operation mode allows the PLL to be enabled or disabled depending on peripheral clock requests.

Note: If ONDEMAND is set, the PLL will only be running when requested by a peripheral and enabled (PLLCTRL.ENABLE = 1). If there is no peripheral requesting the PLL's clock source, the PLL will be in a disabled state. If ONDEMAND is disabled, the PLL will always be running when enabled (PLLCTRL.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active.

This bit is PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE = 1.

Value	Description
0	The PLL0 is always on.
1	The PLL0 is running when a peripheral is requesting the PLL to be used as a clock source. The PLL is not running if no peripheral is requesting the PLL clock source.

Bit 2 – WRTLOCK Write Lock

Note: Once the WRTLOCK bit is set, it can only be cleared by a reset.

Value	Description
0	The PLLCTRL, PLLFBDIV, PLLREFDIV and PLLPOSTDIVA/B registers can be modified by a system write.
1	The PLLCTRL, PLLFBDIV, PLLREFDIV and PLLPOSTDIVA/B registers are write protected, except for bits PLLPOSTDIVA.OUTENn.

Bit 1 – ENABLE PLL Enable

Value	Description
0	The PLL0 oscillator is disabled.
1	The PLL0 oscillator is enabled.

19.7.14 PLL Feed-Back Divider

Name: PLL0FBDIV
Offset: 0x44
Reset: 0x00000000
Property: RW

Table 19-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							FBDIV[9:8]	
Access							RW	RW
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	FBDIV[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – FBDIV[9:0] PLL Feed-Back Divider Factor

This field determines the ratio of the PLL's VCO output frequency to the PLL Reference input frequency. Writing to the FBDIV bits will cause lock to be lost.

The value of FBDIV, (i.e. PLLFBDIV) must be within the range $21 \leq \text{FBDIV} \leq 1023$.

Note: Note: The frequency of the Voltage Controlled Oscillator (VCO) giving the PLL0 oscillation is given by the formula:

$\text{FVCO} = \text{FCKR} * (\text{FBDIV} / \text{REFDIV})$, (i.e., Must be between 800 MHz and 1600 MHz).

f_{CKR} , REFDIV and FBDIV must be selected to satisfy this condition.

19.7.15 PLL reference divider

Name: PLL0REFDIV
Offset: 0x48
Reset: 0x00000000
Property: RW

Table 19-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			REFDIV[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:0 – REFDIV[5:0] PLL reference division factor

This field determines the division factor of the PLL0 input reference frequency. Writing to the REFDIV bits will cause lock to be lost. REFDIV value must be in the range of $1 \leq \text{REFDIV} \leq 63$.

The frequency after the reference divider (F_{PPD}) is given by the formula:

$F_{PPD} = F_{CKR} / \text{REFDIV}$ (i.e., F_{PPD} must always be between 4 MHz to 48 MHz).

19.7.16 PLL output clock divider A

Name: PLL0POSTDIVA
Offset: 0x4C
Reset: 0x20202020
Property: RW



Important: The PLL0 frequency cannot be changed on the fly while it's the active enabled clock to the system.

Table 19-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	OUTEN					POSTDIV[5:0]		
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		1	0	0	0	0	0

Bit 7 – OUTEN PLL output 0 enable

Bits 5:0 – POSTDIV[5:0] PLL output 0 clock division factor

19.7.17 PLL output clock divider B

Name: PLL0POSTDIVB
Offset: 0x50
Reset: 0x00000020
Property: RW



Important: The PLL0 frequency cannot be changed on the fly while it's the active enabled clock to the system.

Table 19-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	OUTEN					POSTDIV[5:0]		
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		1	0	0	0	0	0

Bit 7 – OUTEN PLL output 4 enable

Bits 5:0 – POSTDIV[5:0] PLL output 4 clock division factor

19.7.18 PLL Control

Name: PLL1CTRL
Offset: 0x54
Reset: 0x00000000
Property: RW

Table 19-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			BWSEL[2:0]			REFSEL[2:0]		
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ONDEMAND					WRTLOCK	ENABLE	
Access	RW					RW	RW	
Reset	0					0	0	

Bits 13:11 – BWSEL[2:0] Bandwidth selection

Value	Name	Description
0x0	BWSEL0	TBD
0x1	BWSEL1	TBD
0x2	BWSEL2	TBD
0x3	BWSEL3	TBD
0x4	BWSEL4	TBD
0x5	BWSEL5	TBD
0x6	BWSEL6	TBD
0x7	BWSEL7	TBD

Bits 10:8 – REFSEL[2:0] Reference selection

Value	Name	Description
0x0	GCLK	Dedicated GCLK clock reference
0x1	XOSC	XOSC clock reference
0x2	DFLL48M	DFLL48M clock reference

Bit 7 – ONDEMAND On Demand Control

Bit 2 – WRTLOCK Write Lock

Bit 1 – ENABLE PLL Enable

19.7.19 PLL Feed-Back Divider

Name: PLL1FBDIV
Offset: 0x58
Reset: 0x00000000
Property: RW

Table 19-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							FBDIV[9:8]	
Access							RW	RW
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	FBDIV[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – FBDIV[9:0] PLL Feed-Back Divider Factor

19.7.20 PLL reference divider

Name: PLL1REFDIV
Offset: 0x5C
Reset: 0x00000000
Property: RW

Table 19-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			REFDIV[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:0 – REFDIV[5:0] PLL reference division factor

19.7.21 PLL output clock divider A

Name: PLL1POSTDIVA
Offset: 0x60
Reset: 0x20202020
Property: RW

Table 19-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	OUTEN		POSTDIV[5:0]					
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		1	0	0	0	0	0

Bit 7 – OUTEN PLL output 0 enable

Bits 5:0 – POSTDIV[5:0] PLL output 0 clock division factor

19.7.22 PLL output clock divider B

Name: PLL1POSTDIVB
Offset: 0x64
Reset: 0x00002020
Property: RW

Table 19-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	OUTEN		POSTDIV[5:0]					
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		1	0	0	0	0	0

Bit 7 – OUTEN PLL output 4 enable

Bits 5:0 – POSTDIV[5:0] PLL output 4 clock division factor

19.7.23 Fractional Divider

Name: FRACDIV0
Offset: 0x6C
Reset: 0x00200000
Property: RW

Table 19-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	RW							
Reset	0							

Bits 30:16 – INTDIV[14:0] Frequency division factor integer part

Bits 15:7 – REMDIV[8:0] Frequency division factor reminder part

19.7.24 Fractional Divider

Name: FRACDIV1
Offset: 0x74
Reset: 0x00200000
Property: RW

Table 19-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	RW							
Reset	0							

Bits 30:16 – INTDIV[14:0] Frequency division factor integer part

Bits 15:7 – REMDIV[8:0] Frequency division factor reminder part

19.7.25 Synchronization Busy

Name: SYNCBUSY
Offset: 0x78
Reset: 0x00000000
Property: R

Table 19-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	FRACDIV1	FRACDIV0	DFLLMUL	DFLLDIFF	DFLLTUNE	DFLLCTRLB	DFLLENABLE	
Access	R	R	R	R	R	R	R	
Reset	0	0	0	0	0	0	0	

Bit 7 – FRACDIV1 FRACDIV1 Synchronization Busy

Bit 6 – FRACDIV0 FRACDIV0 Synchronization Busy

Bit 5 – DFLLMUL DFLLMUL Synchronization Busy

This bit is cleared when the synchronization of DFLLMUL register between the clock domains is complete. This bit is set when the synchronization of DFLLMUL register between clock domains is started.

Note: The DFLLMUL synchronization only applies for write operations.

Bit 4 – DFLLDIFF DFLLDIFF Synchronization Busy

This bit is cleared when the synchronization of the DFLLDIFF register between the clock domains is complete. This bit is set when the synchronization of the DFLLDIFF register between clock domains is started.

Note: The DFLLDIFF synchronization only applies for read operations.

Bit 3 – DFLLTUNE DFLLTUNE Synchronization Busy

This bit is cleared when the synchronization of the DFLLTUNE register between the clock domains is complete. This bit is set when the synchronization of the DFLLTUNE register between clock domains is started.

Note: The DFLLTUNE synchronization applies for read and write operations.

Bit 2 – DFLLCTRLB DFLLCTRLB Synchronization Busy

This bit is cleared when the synchronization of the DFLLCTRLB register between the clock domains is complete. This bit is set when the synchronization of the DFLLCTRLB register between clock domains is started.

Note: The DFLLCTRLB synchronization only applies for write operations.

Bit 1 – DFLLENABLE DFLL48M ENABLE Synchronization Busy

This bit is cleared when the synchronization of the DFLLCTRLA.ENABLE register bit between the clock domains is complete.

Note: This bit is set when the synchronization of the DFLLCTRLA.ENABLE register bit between clock domains is started.

19.7.26 XOSC Calibration Register

Name: XOSCCAL
Offset: 0x80
Reset: 0x00000000
Property: RW

Table 19-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	CAL[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CAL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CAL[15:0] XOSC Calibration

19.7.27 RC48M Calibration 0

Name: RC48MCAL0
Offset: 0x88
Reset: 0x00000000
Property: RW

Table 19-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CAL[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CAL[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CAL[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CAL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CAL[31:0] RC48M Calibration

20. Generic Clock Controller (GCLK)

20.1 Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller (GCLK) features 12 Generic Clock Generators that can provide a wide range of clock frequencies.

Generators can be set to use different external or internal oscillators as source. The clock of each Generator can be divided according to the user's requirements. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules. The number of Peripheral Clocks, GCLK_PERIPH, depends on how many peripherals the device has.

Note: Generator 0 is always the direct source of the GCLK_MAIN signal.

20.2 Features

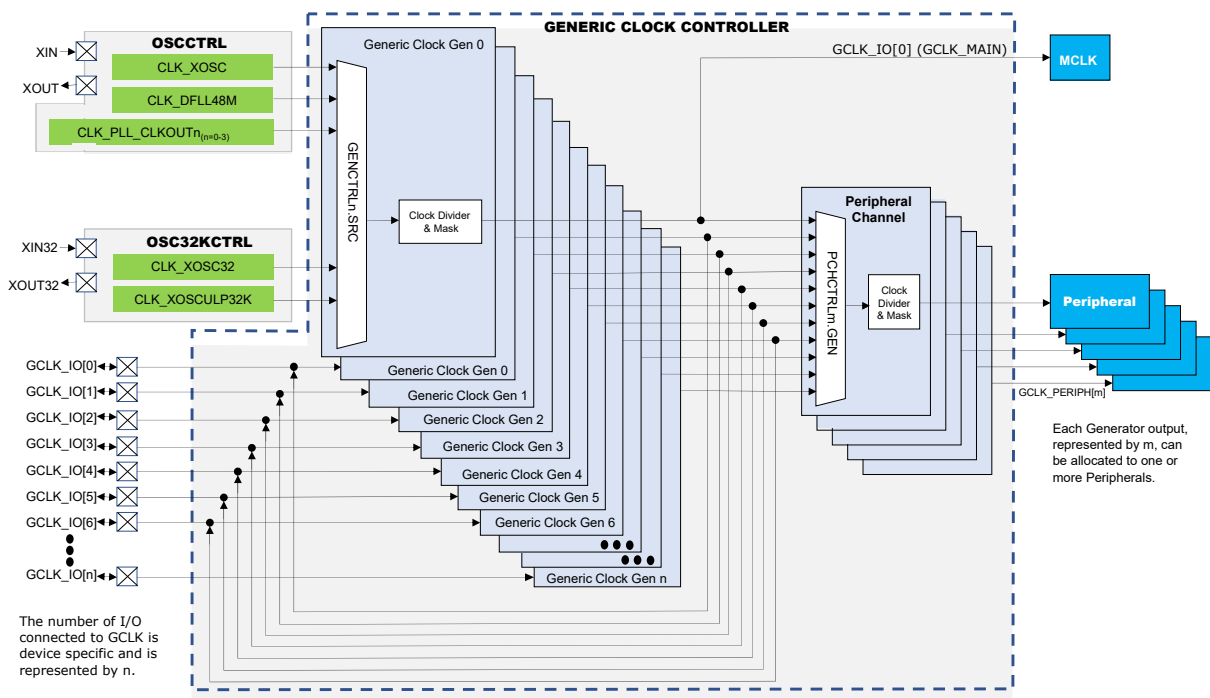
The following are key features of the GCLK module:

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range:
 - Various clock sources
 - Embedded dividers

20.3 Block Diagram

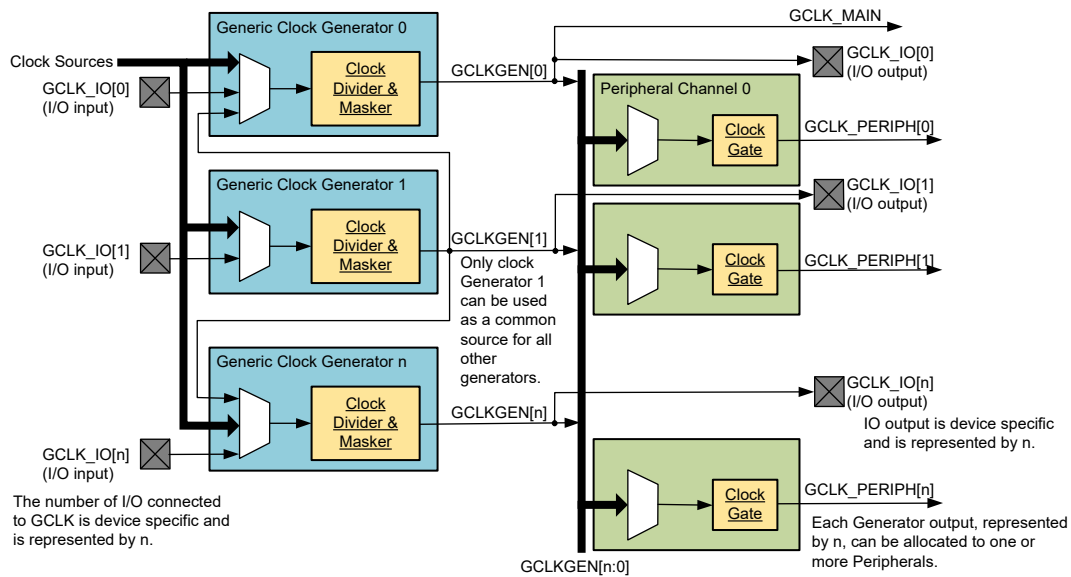
The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in the Device Clocking Diagram below.

Figure 20-1. Device Clocking Diagram



The GCLK block diagram is shown below:

Figure 20-2. Generic Clock Controller Detailed Block Diagram



20.4 Signal Description

Table 20-1. GCLK External Signal Descriptions

Signal Name	Type	Description
GCLK_IO[nseven:0]	Digital I/O	- External Clock source for Generators when input - Generic Clock signal when output

Notes:

- One signal can be mapped on several pins.
- The number of I/O connected to GCLK is device specific. The number is 8 for this design.

20.5 Functional Description

20.5.1 Principle of Operation

The GCLK module is comprised of 12 Generic Clock Generators (Generators) sourcing the Peripheral Channels and the Main Clock signal CLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH) to the peripherals.

20.5.2 GCLK Basic Operation

20.5.2.1 Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

- The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENCTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).

- The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

Note: Each Generator "n" is configured by one dedicated register GENCTRLn.

Note: Each Peripheral Channel "m" is configured by one dedicated register PCHCTRLm.

20.5.2.2 Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to [Configuration Lock](#).

20.5.2.3 Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of 11 different clock sources, GENCTRLn.SRC, except GCLK_GEN[1], which can be set to run from one of seven sources. GCLK_GEN1 is the only Generator that can be selected as source to others Generators.

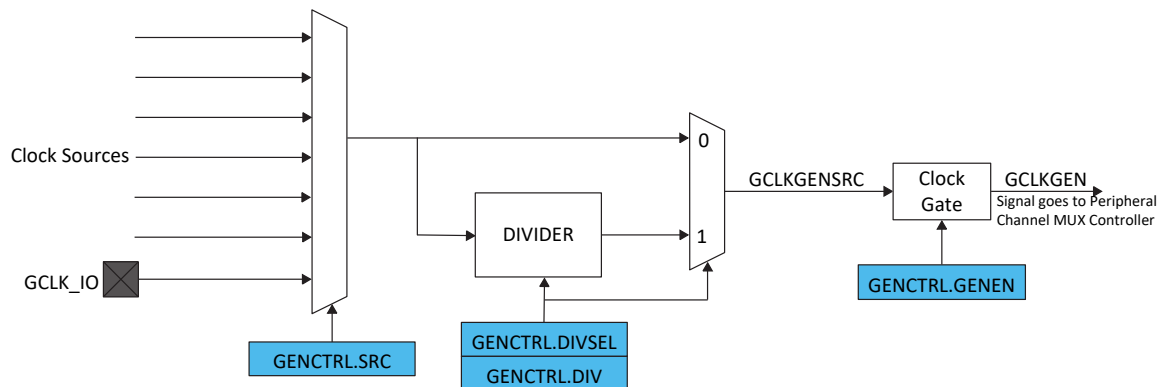
Each Generator GCLK_GEN[x] can be connected to only one specific pin GCLK_IO[x] in a one-to-one GCLK_GENx to GCLK_IOx relationship. A pin GCLK_IO[x] can be set either to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as the clock source for Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller section for details on synchronous clock generation.

Figure 20-3. Generic Clock Generator



20.5.2.4 Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

20.5.2.5 Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated off.

20.5.2.6 Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. Accordingly the SYNCBUSY.GENCTRLn bit in SYNCBUSY register will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, PLL). Only clock Generator 1 can be used as a common source for all other generators.

20.5.2.7 Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

Note: The GCLK Generator source input clock, defined by GENCTRLx.SRC, can be divided as defined in GENCTRLx.DIV.

20.5.2.8 Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

20.5.2.9 External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GEN clock is output to an I/O pin.

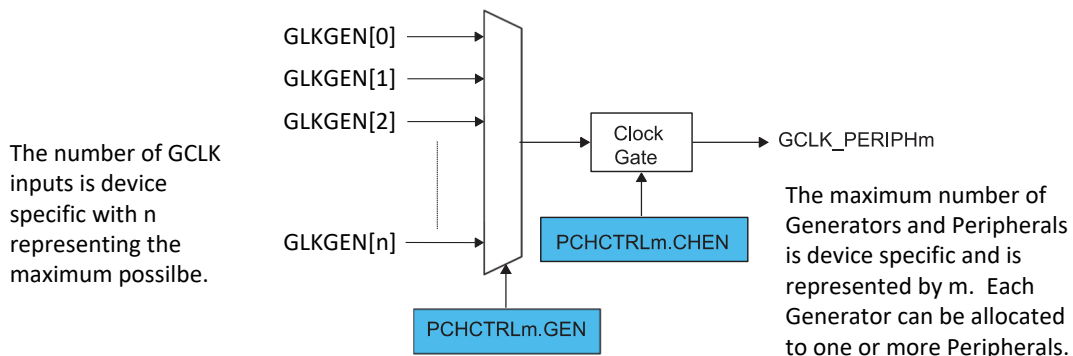
Note: The I/O pin (GCLK/IOn) must first be configured as output by writing the corresponding PORT registers.

If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV. If GENCTRLn.OOV is '0', the output clock will be low. If this bit is '1', the output clock will be high.

In Standby mode, if the clock is output (GENCTRLn.OE = 1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero. If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

20.5.3 Peripheral Clock

Figure 20-4. Peripheral Clock



20.5.3.1 Enabling a Peripheral Clock

Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRLm.GEN). Any available Generator can be selected as clock source for each Peripheral Channel.

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

20.5.3.2 Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN = 0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

20.5.3.3 Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN = 0.
2. Assert that PCHCTRLm.CHEN reads '0'.
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN.
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN = 1.
5. Assert that PCHCTRLm.CHEN reads as '1'.

20.5.3.4 Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too. The corresponding GENCTRLn register is locked, and can be unlocked only by a power reset.

The Generator 0 is one exception concerning the configuration lock. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset, the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again

20.5.4 GCLK Additional Features

20.5.4.1 Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

20.5.5 Sleep Mode Operation

20.5.5.1 SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must first request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during Standby Sleep mode. If the bit is cleared, the Generator output is not available on that pin. When set, the GCLK can continuously output the generator output to GCLK_IO. Refer to [External Clock](#) for details.

20.5.5.2 Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby mode, minimizing the power consumption:

Table 20-2. Clock Generator n Activity in Standby Mode

Request for Clock n present	GENCTRLn.RUNSTDBY	GENCTRLn.OE	Clock Generator n
yes	-	-	active
no	1	1	active
no	1	0	OFF
no	0	1	OFF
no	0	0	OFF

20.5.5.3 Entering Standby Mode

There may occur a delay when the device is put into Standby, until all clocks are gated off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned off properly. The duration of this verification is frequency-dependent.

For further information, refer to the *Power Manager* chapter.

20.5.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers must be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the

synchronization is complete (polling method) and to assert glitch free internal operation. Changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- The Generic Clock Generator Control register (GENCTRLn)
- The Control A register (CTRLA)

Required write synchronization is denoted by the “Write Synchronized” property in the register description.

20.6 Peripheral Dependencies

Table 20-3. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
GCLK	0x44008000		APB: CLKMSK[0] MASK8	---	8 STATUS0 PERID8			PD_CORE_SW

20.7 GCLK Peripheral Channel Control Mapping

Table 20-4. GCLK Peripheral Channel Control Mapping

Peripheral Channel Control Register	Peripheral:Clock
PCHCTRL[0]	OSCCTRL:DFLL48M
PCHCTRL[1]	OSCCTRL:PLL
PCHCTRL[2]	FREQM:MSR0
PCHCTRL[3]	FREQM:MSR1
PCHCTRL[4]	FREQM:REF
PCHCTRL[5]	EIC
PCHCTRL[6]	CAN0
PCHCTRL[7]	CAN1
PCHCTRL[8]	EVSYS:CH0
PCHCTRL[9]	EVSYS:CH1
PCHCTRL[10]	EVSYS:CH2
PCHCTRL[11]	EVSYS:CH3
PCHCTRL[12]	EVSYS:CH4
PCHCTRL[13]	EVSYS:CH5
PCHCTRL[14]	EVSYS:CH6
PCHCTRL[15]	EVSYS:CH7
PCHCTRL[16]	EVSYS:CH8
PCHCTRL[17]	EVSYS:CH9
PCHCTRL[18]	EVSYS:CH10
PCHCTRL[19]	EVSYS:CH11

Table 20-4. GCLK Peripheral Channel Control Mapping (continued)

Peripheral Channel Control Register	Peripheral:Clock
PCHCTRL[20]	SERCOM0:SLOW SERCOM1:SLOW SERCOM2:SLOW SERCOM3:SLOW
PCHCTRL[21]	SERCOM0:CORE
PCHCTRL[22]	SERCOM1:CORE
PCHCTRL[23]	SERCOM2:CORE
PCHCTRL[24]	SERCOM3:CORE
PCHCTRL[25]	TCC0:TCC01 TCC1:TCC01
PCHCTRL[26]	TCC2
PCHCTRL[27]	TCC3
PCHCTRL[28]	SERCOM4:CORE
PCHCTRL[29]	SERCOM5:CORE
PCHCTRL[30]	TCC4
PCHCTRL[31]	TCC5
PCHCTRL[32]	TCC6
PCHCTRL[33]	ADC
PCHCTRL[34]	AC
PCHCTRL[35]	PTC
PCHCTRL[36]	CCL0
PCHCTRL[37]	CCL1
PCHCTRL[38]	USB

20.8 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01	Reserved									
...										
0x03										
0x04	SYNCBUSY	31:24							GENCTRL[15:14]	
		23:16								
		15:8	GENCTRL[13:6]							
		7:0	GENCTRL[5:0]							SWRST
0x08	Reserved									
...										
0x1F										
0x20	GENCTRL0	31:24	DIV[15:8]							
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0				SRC[4:0]				
...										
0x5C	GENCTRL15	31:24	DIV[15:8]							
		23:16	DIV[7:0]							
		15:8			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
		7:0				SRC[4:0]				
0x60	Reserved									
...										
0x7F										
0x80	PCHCTRL	31:24								
		23:16								
		15:8								
		7:0	WRTLOCK	CHEN			GEN[3:0]			

20.8.1 Control

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: RW

Table 20-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								RW
Reset								0

Bit 0 – SWRST Software Reset

Writing a zero to this bit has no effect.

Setting this bit to 1 will reset all registers in the GCLK to their initial state after any reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.

Refer to GENCTRL Reset Value for details on GENCTRL register reset.

Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.

Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Value	Description
0	There is no Reset operation ongoing.
1	A Reset operation is ongoing.

20.8.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: R

Table 20-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
							GENCTRL[15:14]	
Access							R	R
Reset							0	0
Bit	15	14	13	12	11	10	9	8
	GENCTRL[13:6]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	GENCTRL[5:0]							SWRST
Access	R	R	R	R	R	R		R
Reset	0	0	0	0	0	0		0

Bits 17:2 – GENCTRL[15:0] Generic Clock Generator Control x Synchronization Busy bit

Bit 0 – SWRST Software Reset Synchronization Busy bit

This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.

20.8.3 Generic Clock Generator Control

Name: GENCTRL
Offset: 0x20 + n*0x04 [n=0..15]
Reset: 0x00000000
Property: RW

Table 20-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIV[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DIV[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
			RUNSTDBY	DIVSEL	OE	OOV	IDC	GENEN
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				SRC[4:0]				
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bits 31:16 – DIV[15:0] Division Factor

These bits represent a division value for the corresponding GCLK Generator input source clock defined by GENCTRLn.SRC. The actual division factor is dependent on the state of DIVSEL.

Table 20-11. Division Factor Bits

Generic Clock Generator	Division Factor Bits
Clock Generator [15:0]	8 division factor bits - DIV[7:0]

Notes:

If GENCTRLn.DIVSEL = 0 then FGCLK = GENCTRLn.SRC/DIV

- If GENCTRLn.DIV is an odd number, then GENCTRLn.IDC should be set to IDC = 1
- If GENCTRLn.DIV is an even number, then GENCTRLn.IDC should be set to IDC = 0

If GENCTRLn.DIVSEL = 1 then FGCLK = GENCTRLn.SRC/2^(DIV+1)

- GENCTRLn.IDC should ALWAYS be set to IDC = 0

Bit 13 – RUNSTDBY Run in Standby

This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IO pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

Value	Description
0	The Generator is stopped in Standby and the GCLK_IO pin state (one or zero) will be dependent on the setting in GENCTRL.OOV.
1	The Generator is kept running and output to its dedicated GCLK_IO pin during Standby mode.

Bit 12 – DIVSEL Divide Selection

This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.

Value	Name	Description
0	DIV1	Divide input directly by divider factor
1	DIV2	Divide input by $2^{(divider\ factor + 1)}$

Bit 11 – OE Output Enable

This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	No Generator clock signal on pin GCLK_IO.
1	The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO is selected as a generator source in the GENCTRLn.SRC bit field.

Bit 10 – OOV Output Off Value

This bit is used to control the clock output value on pin (GCLK_IO) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

Value	Description
0	The GCLK_IO will be LOW when generator is turned off or when the OE bit is zero.
1	The GCLK_IO will be HIGH when generator is turned off or when the OE bit is zero.

Bit 9 – IDC Improve Duty Cycle

This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

Note: If DIVSEL = 1 this bit should always be set to IDC = 0.

If DIVSEL = 0 and DIV = odd number then IDC=1, else if DIV = even number IDC = 0.

Value	Description
0	Generator output clock duty cycle is not balanced to 50/50 for odd division factors.
1	Generator output clock duty cycle is 50/50.

Bit 8 – GENEN Generic Clock Generator Enable

This bit is used to enable and disable the Generator.

Value	Description
0	Generator is disabled.
1	Generator is enabled.

Bits 4:0 – SRC[4:0] Source Select

These bits select the Generator clock source, as shown in this table.

Table 20-8. Generator Clock Source Selection

Value (GENCTRLn.SRC)	Name	Description
0x00	XOSC	XOSC Crystal/Clock Oscillator
0x01	GCLK_IOn	Generator I/O input pin (GCLK_IOn) where n corresponds to the Clock Generator number.

Table 20-8. Generator Clock Source Selection (continued)

Value (GENCTRLn.SRC)	Name	Description
0x02	GCLK_GEN1	Generic clock generator 1 (GCLK1)
0x03	OSCULP32K (32.768 KHz)	Internal Ultra-Low Power 32K RC Oscillator
0x04	XOSC32K (32.768 KHz)	32 KHz Crystal Oscillator
0x05	DPLL48M (RC48M)	Internal 48 MHz RC Oscillator
0x06	PLL0_CLKOUT1	Digital Phase Lock Loop, PLL0 Output 1
0x07	PLL0_CLKOUT2	Digital Phase Lock Loop, PLL0 Output 2
0x08	PLL0_CLKOUT3	Digital Phase Lock Loop, PLL0 Output 3
0x09	PLL0_CLKOUT4	Digital Phase Lock Loop, PLL0 Output 4
0x0A	PLL0_CLKOUT5	Digital Phase Lock Loop, PLL0 Output 5
0x0B-0x1F	Reserved	Reserved

Note: GENCTRL1.SRC = 0x2 is invalid for GCLK_GEN1 only.

Any reset will reset all GENCTRLn registers. The Reset values of the GENCTRLn registers are shown in table below.

Table 20-9. GENCTRLn Reset Value after a Power Reset

GCLK Generator	Reset Value after a Power Reset
GCLK0 (GENCTRL0)	GCLK.GENCTRL0 = 0x00000105 (DPLL48M, Internal 48 MHz RC Oscillator, GCLK0 Enabled)
GCLK1 (GENCTRL1) - GCLK11 (GENCTRL11)	GCLK.GENCTRL1- GCLK.GENCTRL11 = 0x00000000 (XOSC, GCLK1-GCLK11 disabled)

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK = 1). The reset values of the GENCTRL register are as shown in the table below.

Table 20-10. GENCTRLn Reset Value after a User Reset

GCLK Generator	Reset Value after a User Reset
GCLK0 (GENCTRL0)	GCLK.GENCTRL0 = 0x00000105 (DPLL48M, Internal 48MHz RC Oscillator, GCLK0 Enabled)
GCLK1 (GENCTRL1) - GCLK11 (GENCTRL11)	GCLK.GENCTRL1- GCLK.GENCTRL11 = No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK = 1 else 0x00000000

20.8.4 Peripheral Clock Control

Name: PCHCTRL
Offset: 0x80
Reset: 0x00000000
Property: RW

PCHCTRLm controls the settings of Peripheral Channel number m (m=0..38).

Refer to the Table for finding out the m value corresponding to each module input clock(s).

Table 20-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	WRTLOCK	CHEN			GEN[3:0]			
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0

Bit 7 – WRTLOCK Write Lock

After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.

Note that Generator 0 cannot be locked.

Value	Description
0	The Peripheral Channel register and the associated Generator register are not locked
1	The Peripheral Channel register and the associated Generator register are locked

Bit 6 – CHEN Channel Enable

This bit is used to enable and disable a Peripheral Channel.

Value	Description
0	The Peripheral Channel is disabled
1	The Peripheral Channel is enabled

Bits 3:0 – GEN[3:0] Generic Clock Generator

This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

Table 20-13. Generator Selection

Value	Description
0x0	Generic Clock Generator 0
0x1	Generic Clock Generator 1
0x2	Generic Clock Generator 2
0x3	Generic Clock Generator 3
0x4	Generic Clock Generator 4
0x5	Generic Clock Generator 5
0x6	Generic Clock Generator 6
0x7	Generic Clock Generator 7
0x8	Generic Clock Generator 8
0x9	Generic Clock Generator 9
0xA	Generic Clock Generator 10
0xB	Generic Clock Generator 11
0xC - 0xF	Reserved

Table 20-14. Reset Value after a User Reset or a Power Reset

Reset	PCHCTRLm.GEN	PCHCTRLm.CHEN	PCHCTRLm.WRTLOCK
Power Reset	0x0	0x0	0x0
User Reset	0x0	0x0	0x0

A Power Reset will reset all the PCHCTRLm registers.

A User Reset will reset a PCHCTRL if WRTLOCK=0, or else the content of that PCHCTRL remains unchanged.

The PCHCTRL register Reset values are shown in the table below, PCHCTRLm Mapping.

Value	Name	Description
0x0	GCLK0	Generic clock generator 0
0x1	GCLK1	Generic clock generator 1
0x2	GCLK2	Generic clock generator 2
0x3	GCLK3	Generic clock generator 3
0x4	GCLK4	Generic clock generator 4
0x5	GCLK5	Generic clock generator 5
0x6	GCLK6	Generic clock generator 6
0x7	GCLK7	Generic clock generator 7
0x8	GCLK8	Generic clock generator 8
0x9	GCLK9	Generic clock generator 9
0xA	GCLK10	Generic clock generator 10
0xB	GCLK11	Generic clock generator 11
0xC	GCLK12	Generic clock generator 12
0xD	GCLK13	Generic clock generator 13
0xE	GCLK14	Generic clock generator 14
0xF	GCLK15	Generic clock generator 15

21. Main Clock (MCLK)

21.1 Overview

The Main Clock (MCLK) controls the generation of the device synchronous clocks.

Using a clock provided by the Generic Clock Generator 0 Module (GCLK_MAIN), the MCLK provides synchronous clocks to the CPU, the three Advanced Peripheral Buses (APB), the Advanced High-performance Buses (AHB), and the peripherals connected to these buses.

The synchronous system clocks are divided into 3 clock domains, each can run at different frequencies. The generated clocks can be masked for individual modules, enabling the user to minimize power consumption.

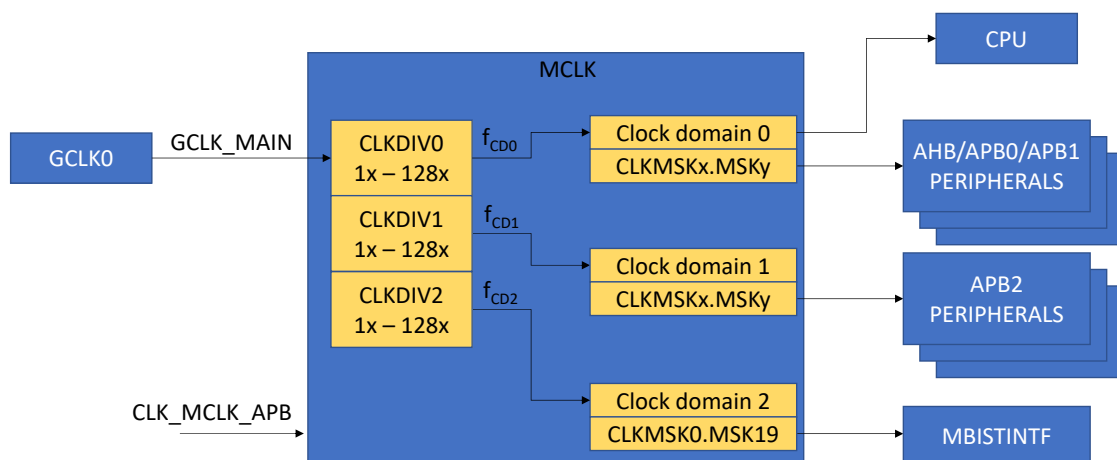
21.2 Features

- **Clock source:** GCLK_MAIN derived from GCLK0
- Three clock domains:
 - 1, 2, 4, 8, 16, 32, 64, 128 independent division factors from GCLK_MAIN
 - **Clock domain 0:** Generates individual clocks running at f_{CD0} frequency for CPU, AHB, APB0/1, and all peripherals connected to them⁽¹⁾
 - **Clock domain 1:** Generates individual clocks running at f_{CD1} frequency for APB2 and all peripherals connected to it⁽¹⁾
 - **Clock domain 2:** Generates a clock running at f_{CD2} dedicated for the MBISTINTF module⁽¹⁾
- Clock domain divider update on the fly
- Peripheral-level clock gating through individual mask bits

Note:

1. Refer to the [Peripherals Configuration Summary](#) for the list of peripherals and the associated clock domain, and clock mask register and index.

21.3 Block Diagram



21.4 Functional Description

21.4.1 Principle of Operation

The GCLK_MAIN clock signal from the GCLK0 module is the source for the MCLK, which in turn is the common root for the synchronous clocks for the CPU, APB0/1/2 (Advanced Peripheral Bus), and the AHB (Advanced High-performance Bus).

For each clock domain, the CLKDIVx.DIV bitfield allows selecting a power of 2 divided clock (from 1 to 128) from the GCLK_MAIN clock, ensuring synchronous clock sources for each clock domain. The default configuration after reset and ROM code execution depends on the user configuration of BOOTCFG1/1A.MCLKDIVx registers.

21.4.2 Basic Operation

21.4.2.1 Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

21.4.2.2 Selecting the Synchronous Clock Division Ratio

Each CLKDIVx.DIV divider can be changed on the fly without halting or disabling the peripheral modules connected to it. This change allows responding to variable load in the application and is applied to all synchronous clocks belonging to the corresponding clock domain at the same time.

To ensure correct operation:

- Frequencies must be selected so that $fCD0 \geq fCD1 \geq fCD2$. Values that violates this relationship will be discarded and will generate a bus error, logged in the PAC module.
- Frequencies must never exceed the specified maximum frequency for each clock domain given in the electrical characteristics specifications.
- Wait for INTFLAG.CLKRDY = 1, change CLKDIVx.DIV, and wait for CLKRDY to come back to 1.

21.4.2.3 Clock Ready Flag

There is a slight delay between writing to CLKDIVx.DIV until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective.

21.4.2.4 Peripheral Clock Masking

All synchronous clocks can then be masked/unmasked individually by writing the corresponding bit in the Clock Mask registers (CLKMSKx.MASKy), where the x and y indexes are defined in the [Peripherals Configuration Summary](#).

All synchronous clocks are unmasked by default after reset.

When a peripheral clock is unmasked and the peripheral is:

- Not enabled (CTRLA.ENABLE = 0), then the clock is automatically gated at the MCLK output to save power consumption, and automatically ungated when read/write register accesses are made in this peripheral
- Enabled (CTRLA.ENABLE = 1), the clock is permanently requested and delivered

When a peripheral clock is masked, its synchronous clock is gated at the MCLK level, saving power consumption in the clock tree, but making the peripheral registers no more accessible. Read/write accesses to the peripheral registers will fail and return an error.

Clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash Memory. Switching off the clock to the MCLK module (which contains the mask registers) or the APB0 bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

The CPU clock to the Arm Cortex processor is derived from the clock domains 0 and runs at fCD0. It is controlled automatically and does not depend on a CLKMSKx.MASK bit.

21.4.3 Interrupts

The peripheral has the following interrupt sources:

- **Clock Ready (CKRDY):** indicates that the programmed divider setting in any CLKDIVx.DIV has taken effect. This interrupt is a synchronous wake-up source.

21.4.4 Sleep Mode Operation

In Active, Idle, and Sleepwalking modes, the MCLK is will automatically gate any clock domain where all peripherals are not requesting their clocks.

In Standby Sleep mode, the MCLK is frozen if no synchronous clock is required.

21.4.5 Debug Operation

When the CPU is halted in Debug mode, the MCLK continues normal operation. In Sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in Debug mode.

21.5 Peripheral Dependencies

Table 21-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
MCLK	0x4400A000	9 : CKRDY	APB: CLKMSK[0] MASK9	---	9 STATUS0 PERID9			PD_CORE_SW

21.6 MCLK AHB/APB Clock Enable Peripheral Mapping

Table 21-2. MCLK AHB/APB Clock Enable Peripheral Mapping

Peripheral BUS Clock Enable Mask Register	Clock Enable Mask Bits	Peripheral	Bus
CLKMSK[0]	MASK0	DSU	AHB
	MASK1	FCR	AHB
	MASK2	FCW	AHB
	MASK3	PM	APB
	MASK4	SUPC	APB
	MASK5	RSTC	APB
	MASK6	OSCCTRL	APB
	MASK7	OSC32KCTRL	APB
	MASK8	GCLK	APB
	MASK9	MCLK	APB
	MASK10	FREQM	APB
	MASK11	WDT	APB
	MASK12	RTC	APB
	MASK13	EIC	APB
	MASK14	PAC	APB
	MASK15	TRAM	APB
	MASK21	MCRAMC	APB
	MASK22	MCRAMC	AHB
	MASK23	CAN0	AHB
	MASK24	CAN1	AHB
	MASK25	H2PB0	AHB
	MASK26	H2PB0	APB
	MASK27	PORT	APB
	MASK28	DMAC	AHB
	MASK29	DMAC	APB
	MASK30	HMATRIX2	AHB
	MASK31	HMATRIX2	APB

Table 21-2. MCLK AHB/APB Clock Enable Peripheral Mapping (continued)

Peripheral BUS Clock Enable Mask Register	Clock Enable Mask Bits	Peripheral	Bus
CLKMSK[1]	MASK0	BROMC	AHB
	MASK1	BROMC	APB
	MASK2	IDAU	APB
	MASK3	EVSYS	APB
	MASK4	SERCOM0	APB
	MASK5	SERCOM1	APB
	MASK6	SERCOM2	APB
	MASK7	SERCOM3	APB
	MASK8	TCC0	APB
	MASK9	TCC1	APB
	MASK10	TCC2	APB
	MASK11	TCC3	APB
	MASK12	SERCOM4	APB
	MASK13	SERCOM5	APB
	MASK14	TCC4	APB
	MASK15	TCC5	APB
	MASK16	TCC6	APB
	MASK17	ADC	APB
	MASK18	AC	APB
	MASK19	PTC	APB
	MASK20	CCL0	APB
	MASK21	CCL1	APB
	MASK22	USB	AHB
	MASK23	USB	APB
	MASK24	AT	APB
	MASK25	H2PB1	AHB
	MASK26	H2PB1	APB
	MASK27	H2PB2	AHB
	MASK28	H2PB2	APB
	MASK30	HSM	AHB
	MASK31	PUF	APB

21.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								CKRDY
0x04	INTENSET	31:24								
		23:16								
		15:8								
		7:0								CKRDY
0x08	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								CKRDY
0x0C	CLKDIV0	31:24								
		23:16								
		15:8								
		7:0	DIV[7:0]							
0x10 ... 0x3B	Reserved									
0x3C	CLKMSK0	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x40	CLKMSK1	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x44	CLKMSK2	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x48	CLKMSK3	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x4C	CLKMSK4	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x50	CLKMSK5	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x54	CLKMSK6	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x58	CLKMSK7	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
0x5C	CLKMSK8	31:24	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
		23:16	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
		15:8	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
		7:0	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0

21.7.1 Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x0
Reset: 0x00000000
Property: R/K

Table 21-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/K
Reset								0

Bit 0 – CKRDY Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Note: Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

Value	Name	Description
1		The Clock Ready interrupt is enabled.
0		The Clock Ready interrupt is disabled

21.7.2 Interrupt Enable Set Register

Name: INTENSET
Offset: 0x4
Reset: 0x00000000
Property: R/S

Table 21-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								R/S
Reset								0

Bit 0 – CKRDY Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

Value	Name	Description
1		The Clock Ready interrupt is enabled.
0		The Clock Ready interrupt is disabled.

21.7.3 Interrupt Flag Status Register

Name: INTFLAG
Offset: 0x8
Reset: 0x00000001
Property: K

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 21-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								CKRDY
Access								HS/K
Reset								1

Bit 0 – CKRDY This flag is set when the system clocks have frequencies as indicated in the CLKDIVx registers and will generate an interrupt if CKRDY interrupt enable is set to '1'.

Notes:

1. Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.
2. This flag is **set by hardware** when the system clocks have frequencies as indicated in the CLKDIVx registers and will generate an interrupt if CKRDY interrupt enable is set to '1'.

Value	Description
0	The Clock Ready interrupt is disabled.
1	The Clock Ready interrupt is enabled.

21.7.4 Clock Divider Control n Register

Name: CLKDIV
Offset: 0x0C
Reset: 0x00000000
Property: R/W

Notes:

1. the CLKDIV0.DIV bit field is write protected.
2. To ensure correct operation, frequencies must be selected so that $\text{CLKDIV0.DIV} \leq \text{CLKDIV1.DIV} \leq \text{CLKDIV2.DIV}$.
3. Frequencies must never exceed the specified maximum frequency for each clock domain.
4. The user updates to this register may not take effect immediately. The MCLK module logic will wait for the falling edge of the previous clock and the new clock to coincide before switching. The INTFLAG.CKRDY can be used to determine when MCLK has made the switch.

Table 21-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	DIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – DIV[7:0] These bits define the division ratio of the main clock prescaler related to the Clock Domain controlled by the CLKDIVn register. To ensure correct operation, frequencies must be selected so that $\text{FCLKDIV}[n] \leq \text{FCLKDIV}[n+1]$ (i.e. $\text{CLKDIVn.DIV} \leq \text{CLKDIV}(n+1).\text{DIV}$). Frequencies must never exceed the specified maximum frequency for each clock domain. An update to this register may not take effect immediately. The MCLK macro will wait for the falling edge of the previous clock and the new clock to coincide before switching. The INTFLAG.CKRDY can be used to determine when MCLK has made the switch. This field is write protected if $\text{MCLK\_CKDIV\_IS\_WRLOCK}[n]=1$. All other values are reserved.

These bits define the division ratio of the main clock, (MCLK), prescaler related to the CPU Clock Domain controlled by the CLKDIVn register.

Note: All other values are reserved or invalid.

Value	Name	Description
1	DIV1	Divide by 1
2	DIV2	Divide by 2
4	DIV4	Divide by 4
8	DIV8	Divide by 8
16	DIV16	Divide by 16
32	DIV32	Divide by 32
64	DIV64	Divide by 64
128	DIV128	Divide by 128

21.7.5 Peripheral Clock Enable Mask n Register

Name: CLKMSK
Offset: 0x3C + n*0x04 [n=0..8]
Reset: 0x00000000
Property: R/W

Table 21-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MASK31	MASK30	MASK29	MASK28	MASK27	MASK26	MASK25	MASK24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MASK23	MASK22	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – MASK Unimplemented bits become undefined.

Value	Name	Description
0		Peripheral clock mclk_clk_periph[n*32+x] stopped
1		Peripheral clock mclk_clk_periph[n*32+x] enabled

22. 32 KHz Oscillators Controller (OSC32KCTRL)

22.1 Overview

The 32 kHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768 kHz oscillators:

- A 32.768 kHz crystal oscillator (XOSC32K)
- A 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)

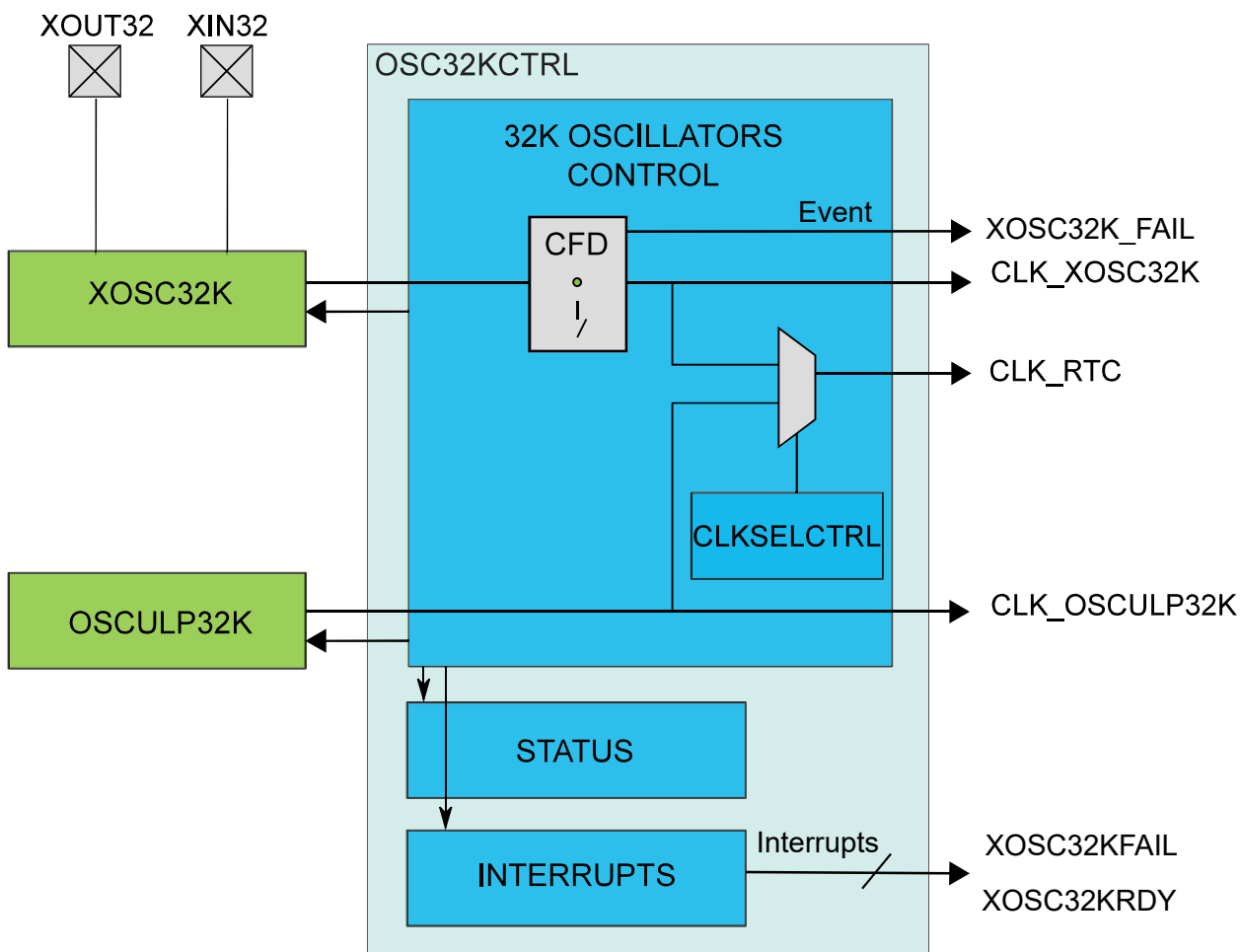
The OSC32KCTRL sub-peripherals can be enabled, disabled, and monitored through interface registers. All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes through the INTENSET, INTENCLR, and INTFLAG registers. The OSC32KCTRL provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), Watchdog Timer (WDT), and Reset Power Management Unit (RPMU).

22.2 Features

- 32.768 kHz Crystal Oscillator (XOSC32K)
 - Programmable start-up time
 - Crystal or external input clock on XIN32 I/O
 - Clock failure detection with safe clock switch
 - Clock failure event output
- 32.768 kHz Ultra Low-Power Internal Oscillator (OSCULP32K)
 - Ultra low-power, always-on oscillator
 - Frequency fine tuning
- 1.024 kHz clock outputs available

22.3 Block Diagram

Figure 22-1. OSC32KCTRL Block Diagram



22.4 Signal Description

Signal	Description	Type
XIN32	Analog Input	32.768 kHz Crystal Oscillator or external clock input
XOUT32	Analog Output	32.768 kHz Crystal Oscillator output

The I/O lines are automatically selected when XOSC32K is enabled and do not need to be configured by the user. The external 32.768 kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors.

22.5 Functional Description

22.5.1 Principle of Operation

The XOSC32K and OSCULP32K are configured through the OSC32KCTRL control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from Standby mode, provided the corresponding interrupt is enabled.

22.5.2 32 kHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in two different modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768 kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode (XOSC32K.XTALEN = 1), the XIN32 and XOUT32 pins are controlled by XOSC32K.ENABLE, and the GPIO functions are overridden on both pins. When in external clock mode (XOSC32K.XTALEN = 0), only the XIN32 pin will be overridden and controlled by XOSC32K.ENABLE, while the XOUT32 pin can still be used as a GPIO pin.

Enabling, Disabling: The XOSC32K is enabled by writing a '1' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 1). It is necessary to wait for STATUS.XOSC32KRDY bit to set, before disabling the XOSC32K. The XOSC32K is disabled by writing a '0' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 0). It is necessary to wait for STATUS.XOSC32KRDY to clear before enabling the XOSC32K.

Mode Selection: To enable the XOSC32K in Crystal Oscillator mode, the XTALEN bit in the 32 kHz External Crystal Oscillator Control register must be written (XOSC32K.XTALEN = 1). If XOSC32K.XTALEN is '0', the External Clock Input mode will be enabled. This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored.

Gain Boost: To boot the oscillator in Crystal mode, the BOOST bit in the 32 kHz External Crystal Oscillator Control register must be written (XOSC32K.BOOST = 1). This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored. If the highest gain settings are not enough, then the module should be put in boost mode to force start an oscillation.

Gain Selection: When a crystal oscillator is selected, a user controllable gain is provided through the CGM control bits. Users should review the electrical specification. This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored.

Startup: XOSC32K.STARTUP[3:0] selects the startup time. It is configurable and applicable for crystal mode or external clock mode. These bits must be written when XOSC32K.ENABLE = 0. Otherwise, the write on these bits is ignored.

The XOSC32K will behave differently in different sleep modes based on the settings of the XOSC32K.ONDEMAND and XOSC32K.ENABLE bits. The XOSC32K.ONDEMAND bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored. If XOSC32K.ENABLE = 0, the XOSC32K will be always stopped. For XOSC32K.ENABLE = 1, following table is valid:

XOSC32K.ONDEMAND	Sleep Behavior
0	Always run
1	Run if requested by a peripheral

As a crystal oscillator usually requires a long start-up time, the 32 kHz External Crystal Oscillator will keep running across resets, except for Power-on Reset (POR). After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP[3:1]) in the 32 kHz External Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY = 1). The transition of

STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY = 1).

22.5.3 Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC32K). The CFD detects failing operation of the XOSC32K clock with reduced latency and allows to switch to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC32K in case of recovery. The safe clock is derived from the OSCULP32K oscillator with a configurable pre-scaler. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral.

The Oscillator's registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can be configured to trigger an interrupt or an event when a failure is detected.

Clock Failure Detection

The CFD is reset only at Power-on Reset (POR). The CFD does not monitor the XOSC32K clock when the oscillator is disabled (XOSC32K.ENABLE = 0).

Before starting CFD operation, the user must start and enable the safe clock source (OSCULP32K oscillator). The CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (CFDCTRL.CFDEN). After starting or restarting the XOSC32K, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up Time in the External Multipurpose Crystal Oscillator Control register (XOSC32K.STARTUP). Once the XOSC32K Start-Up Time is elapsed, the XOSC32K clock is constantly monitored.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC32K. There must be at least one rising and one falling XOSC32K clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.XOSC32KFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.XOSC32KFAIL) are set. If the XOSC32KFAIL bit in the Interrupt Enable Set register (INTENSET.XOSC32KFAIL) is set, an interrupt is generated. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated.

After a clock failure is issued, the CFD continuously monitors of the XOSC32K clock and the Clock Failure Detector status bit in the Status register (STATUS.XOSC32KFAIL) reflects the current XOSC32K activity.

Clock Switch

When a clock failure is detected, the XOSC32K clock is replaced by the safe clock in order to maintain an active clock during the XOSC32K clock failure. The safe clock source is the OSCULP32K oscillator clock. Both 32KHz and 1KHz outputs of the XOSC32K are replaced by the OSCULP32K, 32 kHz and 1 kHz outputs, respectively. The safe clock source can be scaled down by a configurable pre-scaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC32K clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.XOSC32KSW) is set.

When the CFD switches 32 kHz clock from XOSC32K to the safe clock, the XOSC32K is still enabled. If required, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations. In the case the application can recover the XOSC32K, the application can switch back to the XOSC32K clock by writing a '1' to Switch Back Enable bit in the Clock Failure

Control register (CFDCTRL.SWBACK). Once the XOSC32K clock is switched back, the Switch Back bit (CFDCTRL.SWBACK) is cleared by hardware.

Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler allows to scale down the OSCULP32K oscillator clock to half so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD.

The prescaler is applied on both outputs (32 kHz and 1 kHz) of the safe clock.

Note: For an external crystal oscillator at 32 kHz and the OSCULP32K frequency is 32 kHz, the XOSC32K.CFDPRESC must be set to 0 for a safe clock of equal frequency.

Sleep Mode

The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. For further details, refer to the XOSC32K Sleep mode Behavior table. The CFD interrupt can be used to wake up the device from sleep modes.

22.5.4 32 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra low-power clock source. The OSCULP32K is factory-calibrated oscillator under typical voltage and temperature conditions.

The OSCULP32K is enabled by default after a Power-on Reset (POR) and will always run except during POR.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

22.5.5 Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024 kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

22.5.6 RTC Clock Selection Control

Before enabling the system RTC module, the RTC clock must be selected first. All outputs from this oscillator source are valid RTC clock sources. The RTC Clock Selection is available in the Selection Control register (CLKSELCTRL.RTCSEL). To ensure proper operation, it is highly recommended to disable the RTC module before changing its clock source.

22.5.7 Interrupts

The OSC32KCTRL has the following interrupt sources:

- **XOSC32KRDY - 32 kHz Crystal Oscillator Ready:** A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected and XOSC32K ready interrupt request or XOSC32K ready interrupt wake up request are set.
- **XOSC32KFAIL - Clock Failure Detector:** A 0-to-1 transition on the STATUS.XOSC32KFAIL bit is detected and XOSC32K clock fail detector interrupt request or XOSC32K clock fail detector interrupt wake up request are set.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set, and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is

disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL has two interrupt request line each for XOSC32KRDY and XOSC32KFAIL interrupt sources.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

22.5.8 Events

If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output and event line will go high. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

The CFD can generate the following output event:

- Clock Failure Detector (XOSC32KFAIL): Generated when the Clock Failure Detector status bit is set in the Status register (STATUS.XOSC32KFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.XOSC32KSW) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.

22.6 Peripheral Dependencies

Table 22-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
OSC32KCTRL	0x45008000	7 : XOSC32KRDY 8 : XOSC32KFAIL	APB: CLKMSK[0] MASK7	---	7 STATUS0 PERID7		Generators: XOSC32KFAIL: CHANNEL[3]	PD_CORE_BU

22.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8								
		7:0						XOSC32KFAIL		XOSC32KRDY
0x04	INTENSET	31:24								
		23:16								
		15:8								
		7:0						XOSC32KFAIL		XOSC32KRDY
0x08	INTFLAG	31:24								
		23:16								
		15:8								
		7:0						XOSC32KFAIL		XOSC32KRDY
0x0C	STATUS	31:24								
		23:16								
		15:8								
		7:0					XOSC32KSW	XOSC32KFAIL		XOSC32KRDY
0x10	CLKSELCTRL	31:24								
		23:16								
		15:8								
		7:0			HSMSEL[1:0]				RTCSEL[1:0]	
0x14	CFDCTRL	31:24								
		23:16								
		15:8								
		7:0						CFDPRESC	SWBACK	CFDEN
0x18	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								CFDEO
0x1C	XOSC32K	31:24					CTRLX[3:0]			
		23:16			CGM[3:0]				BOOST	ENSL
		15:8					STARTUP[3:0]			
		7:0	ONDEMAND					XTALEN	ENABLE	

22.7.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x00
Reset: 0x00000000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Table 22-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						XOSC32KFAIL		XOSC32KRDY
Access						RW		RW
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detector Interrupt Enable

Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Fail Interrupt Enable bit, (i.e. XOSC32KFAIL), which disables the XOSC32K Clock Failure interrupt
- Writing a one to the same corresponding bit in the INTENSET register

Value	Description
0	The XOSC32K Clock Fail Detect Interrupt is disabled.
1	The XOSC32K Clock Fail Detect Interrupt is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable

Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit, (XOSC32KRDY), will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt

- Writing a one to the same corresponding bit in the INTENSET register

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.7.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x04
Reset: 0x00000000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Table 22-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						XOSC32KFAIL		XOSC32KRDY
Access						RW		RW
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detector Interrupt Enable

Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Fail Interrupt Enable bit, (i.e. XOSC32KFAIL), which disables the XOSC32K Clock Failure interrupt
- Writing a one to the same corresponding bit in the INTENCLR register

Value	Description
0	The XOSC32K Clock Fail Detect Interrupt is disabled.
1	The XOSC32K Clock Fail Detect Interrupt is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set.

Bit 0 – XOSC32KRDY XOSC32K Ready Interrupt Enable

Note: Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit, (XOSC32KRDY), will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt

- Writing a one to the same corresponding bit in the INTENCLR register

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.7.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x08
Reset: 0x00000000
Property: RW

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 22-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						XOSC32KFAIL		XOSC32KRDY
Access						RW		RW
Reset						0		0

Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detector

Note: Writing a '0' to this bit has no effect.

This bit is set by hardware under the following conditions:

- This flag is set by hardware on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the (STATUS.XOSC32KFAIL) register and will generate an Interrupt request if INTENSET.XOSC32KFAIL is '1'

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag

Value	Description
0	No XOSC32K 32.768kHz Clock Fail Detected.
1	XOSC32K 32.768kHz Clock Fail Detected.

Bit 0 – XOSC32KRDY XOSC32K Ready

Note: Writing a '0' to this bit has no effect.

This bit is set by hardware under the following conditions:

- This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY) and will generate an interrupt request if INTENSET.XOSC32KRDY=1

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Ready interrupt flag

Value	Description
0	The XOSC32K Ready interrupt is disabled.
1	The XOSC32K Ready interrupt is enabled.

22.7.4 Power and Clocks Status

Name: STATUS
Offset: 0x0C
Reset: 0x00000000
Property: R

Table 22-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					XOSC32KSW	XOSC32KFAIL		XOSC32KRDY
Access					R	R		R
Reset					0	0		0

Bit 3 – XOSC32KSW XOSC32K Clock switch

Note: This bit is set by hardware if a 32.768kHz clock fail detect occurs and clock fail detect is enabled, (i.e. CFDCTRL.CFDEN =1 and INTFLAG.XOSC32KFAIL=1 or STATUS.XOSC32KFAIL=1).

Value	Description
0	XOSC32K is not switched and the clock source is provided by the external 32.768kHz oscillator source.
1	XOSC32K is switched to the internal OSCULP32K oscillator clock. Both 32.768kHz and 1.024kHz outputs of the XOSC32K are replaced by the respective OSCULP32K 32 kHz and 1 kHz outputs to the respective logic modules.

Bit 2 – XOSC32KFAIL XOSC32K Clock Failure Detector

Note: This bit is set by hardware if a 32.768kHz clock fail detect occurs and clock fail detect is enabled, (i.e. CFDCTRL.CFDEN =1 and INTFLAG.XOSC32KFAIL=1).

Value	Description
0	No XOSC32K 32.768kHz clock fail detection.
1	XOSC32K 32.768kHz external crystal/clock fail detect. Hardware clock switch will be initiated to OSCULP32K oscillator clocks.

Bit 0 – XOSC32KRDY XOSC32K Ready

Note: This bit is set and cleared by hardware based on the status of the of the active XOSC32K 32.768kHz clock source.

Value	Description
0	XOSC32K is not ready.
1	XOSC32K is stable and ready to be used as a clock source.

22.7.5 Clock Selection Control

Name: CLKSELCTRL
Offset: 0x10
Reset: 0x00000000
Property: RW

Table 22-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			HSMSEL[1:0]				RTCSEL[1:0]	
Access			RW	RW			RW	RW
Reset			0	0			0	0

Bits 5:4 – HSMSEL[1:0] HSM Clock Selection

Value	Name	Description
0x0	ULP32K	32.768kHz from 32kHz internal ULP oscillator
0x2	XOSC32K	32.768kHz from 32.768kHz external crystal oscillator

Bits 1:0 – RTCSEL[1:0] RTC Clock Selection

These bits select the RTC clock source.

Value	Name	Description
0x0	ULP32K	32 kHz from 32 kHz internal ULP oscillator
0x1	ULP1K	1 kHz from 32 kHz internal ULP oscillator
0x2	XOSC32K	32.768 kHz from 32 kHz external crystal oscillator
0x3	XOSC1K	1.024 kHz from 32 kHz external oscillator

Value	Name	Description
0x0	ULP32K	32.768kHz from 32kHz internal ULP oscillator
0x1	ULP1K	1.024kHz from 32kHz internal ULP oscillator
0x2	XOSC32K	32.768kHz from 32.768kHz external crystal oscillator
0x3	XOSC1K	1.024kHz from 32.768kHz internal oscillator

22.7.6 Clock Failure Detector Control

Name: CFDCTRL
Offset: 0x14
Reset: 0x00000000
Property: RW

Table 22-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						CFDPRESC	SWBACK	CFDEN
Access						RW	RW	RW
Reset						0	0	0

Bit 2 – CFDPRESC Clock Failure Detector Prescaler

This bit selects the prescaler for the Clock Failure Detector.

Value	Description
0	The CFD safe clock frequency is the OSCULP32K frequency
1	The CFD safe clock frequency is the OSCULP32K frequency divided by 2

Bit 1 – SWBACK Clock Switch Back

This bit controls the XOSC32K output switch back to the external clock or crystal oscillator in case of clock recovery.

Value	Description
0	The clock switch is disabled.
1	The clock switch is enabled. This bit is reset when the XOSC32K output is switched back to the external clock or crystal oscillator.

Bit 0 – CFDEN Clock Failure Detector Enable

This bit selects the Clock Failure Detector state.

Note: After setting CFDEN to enable clock failure detection, STATUS.XOSC32KFAIL will always be set. This first detection must be ignored. Subsequent setting of this bit will indicate actual clock failure events.

Value	Description
0	The CFD is disabled.
1	The CFD is enabled.

22.7.7 Event Control

Name: EVCTRL
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 22-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								CFDEO
Access								RW
Reset								0

Bit 0 – CFDEO Clock Failure Detector Event Output Enable

This bit controls whether the Clock Failure Detector event output is enabled and an event will be generated when the CFD detects a clock failure.

Note: To prevent false event generation, the CFDEO bit must be set or cleared *only* when the XOSC32K is Disabled, (i.e. XOSC32K.ENABLE=0).

Value	Description
0	Clock Failure Detector Event output is disabled, no event will be generated.
1	Clock Failure Detector Event output is enabled, an event will be generated.

22.7.8 32kHz External Crystal Oscillator (XOSC32K) Control

Name: XOSC32K
Offset: 0x1C
Reset: 0x00200080
Property: RW

Table 22-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						CTRLX[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
						CGM[3:0]		BOOST
Access			RW	RW	RW	RW	RW	RW
Reset			1	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						STARTUP[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ONDEMAND					XTALEN	ENABLE	
Access	RW					RW	RW	
Reset	1					0	0	

Bits 27:24 – CTRLX[3:0] Extended Control

Bits 21:18 – CGM[3:0] Control Gain Mode

These bits control the gain of the external crystal oscillator.
These bits are enable-protected.

Value	Name	Description
0x0	CGM0	The lower Control Gain Mode value
0x1	CGM1	A higher Control Gain Mode value than CGM0
0x2	CGM2	A higher Control Gain Mode value than CGM1
0x3	CGM3	A higher Control Gain Mode value than CGM2
0x4	CGM4	A higher Control Gain Mode value than CGM3
0x5	CGM5	A higher Control Gain Mode value than CGM4: • Min Recommended for SF=3, ESR ≤ 100K • Min Recommended for SF=5, ESR ≤ 60K
0x6	CGM6	A higher Control Gain Mode value than CGM5
0x7	CGM7	A higher Control Gain Mode value than CGM6
0x8	CGM8	A higher Control Gain Mode value than CGM7
0x9	CGM9	A higher Control Gain Mode value than CGM8
0xA	CGM10	A higher Control Gain Mode value than CGM9, (Min Recommended for SF=5, ESR ≤ 100K)

CGM (continued)

Value	Name	Description
0xB	CGM11	A higher Control Gain Mode value than CGM10
0xC	CGM12	A higher Control Gain Mode value than CGM11
0xD	CGM13	A higher Control Gain Mode value than CGM12
0xE	CGM14	A higher Control Gain Mode value than CGM13
0xF	CGM15	The highest Control Gain Mode value

Note:

1. These bits are enable-protected. They cannot be written to if XOSC32K.ENABLE = 1.

Value	Name	Description
0x0	CGM0	the lower Control Gain Mode value
0x1	CGM1	a higher Control Gain Mode value than CGM0
0x2	CGM2	a higher Control Gain Mode value than CGM1
0x3	CGM3	a higher Control Gain Mode value than CGM2
0x4	CGM4	a higher Control Gain Mode value than CGM3
0x5	CGM5	a higher Control Gain Mode value than CGM4
0x6	CGM6	a higher Control Gain Mode value than CGM5
0x7	CGM7	a higher Control Gain Mode value than CGM6
0x8	CGM8	a higher Control Gain Mode value than CGM7
0x9	CGM9	a higher Control Gain Mode value than CGM8
0xA	CGM10	a higher Control Gain Mode value than CGM9
0xB	CGM11	a higher Control Gain Mode value than CGM10
0xC	CGM12	a higher Control Gain Mode value than CGM11
0xD	CGM13	a higher Control Gain Mode value than CGM12
0xE	CGM14	a higher Control Gain Mode value than CGM13
0xF	CGM15	the highest Control Gain Mode value

Bit 17 – BOOST Gain Boost

Bit 16 – ENSL Enable Servo Loop

This bit enables the XOSC32K crystal Servo Loop.

Note: This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE=1.

Value	Description
0	The Servo Loop is disabled.
1	The Servo Loop is enabled.

Bits 11:8 – STARTUP[3:0] Startup Mode

These bits select the start-up time for the XOSC32K oscillator. The OSCULP32K oscillator is used to clock the start-up counter. The given time assumes an XOSC32K crystal frequency of 32.768 kHz. After the XOSC32K.STARTUP time has expired, the XOSC32K clock is released internally after the selected programmable startup clock cycles plus 3 additional XOSC32 periods. The Clock Fail Detect (CFD) monitoring also starts when the clock is released for internal use. The user selected start-up time must equal or exceed the start-up time defined in the electrical characteristics.

STARTUP[2:0]	OSCULP32K Clock Cycles	Plus	XOSC32K Clock Cycles	Equal	Approximate Equivalent Time
0x0	1	+	3	=	~122 μ s
0x1	16	+	3	=	~580 μ s
0x2	32	+	3	=	~1.07 ms
0x3	2048	+	3	=	~62.6 ms
0x4	4096	+	3	=	~125.1 ms
0x5	8192	+	3	=	~250.1 ms

STARTUP (continued)

STARTUP[2:0]	OSCULP32K Clock Cycles	Plus	XOSC32K Clock Cycles	Equal	Approximate Equivalent Time
0x6	16384	+	3	=	~500.1 ms
0x7	32768	+	3	=	~1s
0x8	65536	+	3	=	~2s
0x9	131072	+	3	=	~4s
0xA	262144	+	3	=	~8s
0xB – 0xF	---	---	---	---	Reserved

Note: These bits are enable-protected. They cannot be written to if XOSC32K.ENABLE = 1. These bits are valid only when XOSC32K.XTALEN = 1, crystal XOSC32K selected.

Value	Name	Description
0x0	1CYCLE	1 ULP clock cycle
0x1	16CYCLES	16 ULP clock cycles
0x2	32CYCLES	32 ULP clock cycles
0x3	2048CYCLES	2048 ULP clock cycles
0x4	4096CYCLES	4096 ULP clock cycles
0x5	8192CYCLES	8192 ULP clock cycles
0x6	16384CYCLES	16384 ULP clock cycles
0x7	32768CYCLES	32768 ULP clock cycles
0x8	65536CYCLES	65536 ULP clock cycles
0x9	131072CYCLES	131072 ULP clock cycles
0xA	262144CYCLES	262144 ULP clock cycles

Bit 7 – ONDEMAND On Demand Mode

This bit controls how the XOSC32K behaves when a peripheral clock request is detected.

Note: This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE = 1.

Value	Description
0	XOSC32K always run
1	Only run if requested by a peripheral

Bit 2 – XTALEN Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

Note: This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE = 1.

Value	Description
0	External clock connected on XIN32. XOUT32 can be used as general-purpose I/O.
1	Crystal connected to XIN32/XOUT32.

Bit 1 – ENABLE Oscillator Enable

Note: It is necessary to wait for STATUS.XOSC32KRDY = 0 before enabling the XOSC32K.

Value	Description
0	The 32K oscillator is disabled.
1	The 32K oscillator is enabled.

23. Watchdog Timer (WDT)

23.1 Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will cause a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, this will cause a system reset. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

When enabled, the WDT runs in active mode and selectively runs in standby sleep mode. It is asynchronous and runs from a CPU-independent internal clock source. The WDT continues operation and causes a system reset even if the main clocks fail. Even though there can be time between the reset condition occurring and the system reset asserting, it cannot be prevented. This includes but is not limited to a software attempt to disable the WDT.

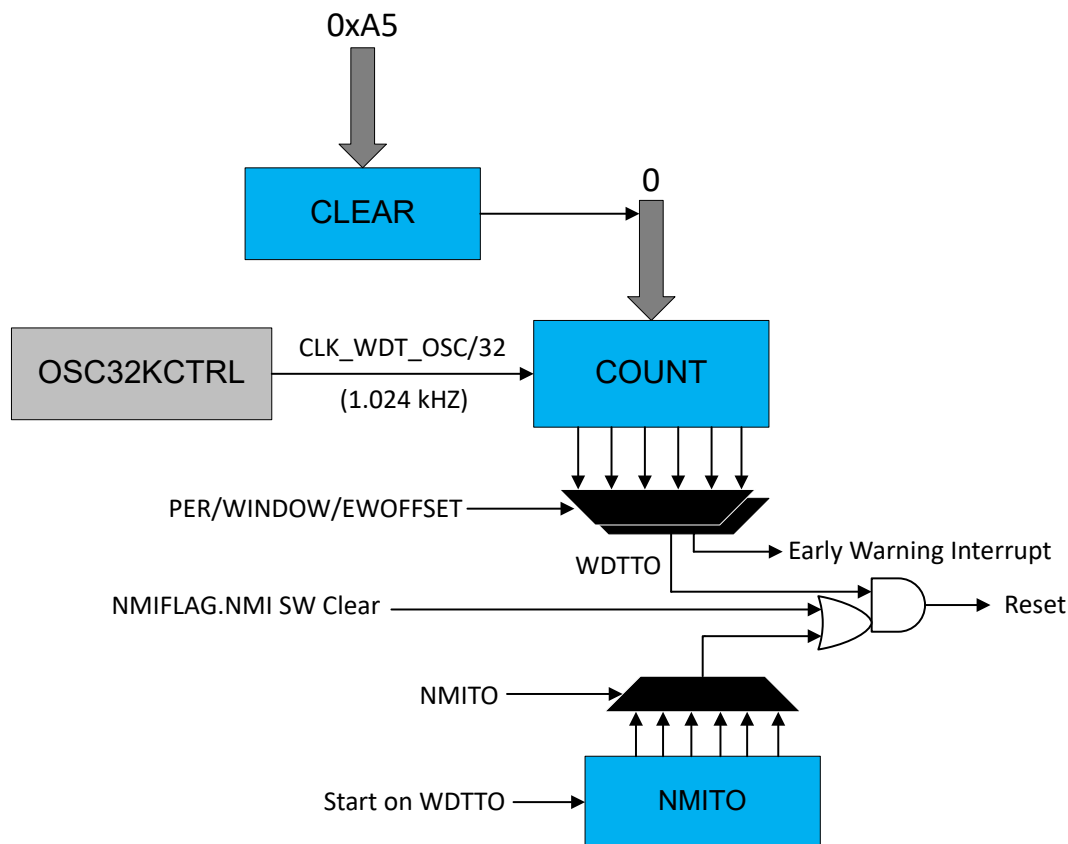
23.2 Features

The following are key features of the WDT module:

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
 - Normal
 - Window mode
- Selectable time-out periods
 - From 8 cycles to 16,384 cycles in Normal mode
 - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

23.3 Block Diagram

Figure 23-1. WDT Block Diagram



23.4 Functional Description

23.4.1 Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

Table 23-1. WDT Operating Modes

CTRLA.ENABLE	CTRLA.WEN	Interrupt Enable	Mode
0	x	x	Stopped
1	0	0	Normal mode
1	0	1	Normal mode with Early Warning interrupt
1	1	0	Window mode
1	1	1	Window mode with Early Warning interrupt

23.4.2 Basic Operation

23.4.2.1 Initialization

The WDT has enable-protected registers which can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined. The Window Enable bit (CTRLB.WEN=1) selects window mode.

When enabling the WDT from the disabled state, the WDT count starts from 0.

23.4.2.2 Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the *USER CFG page*.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Run In Standby Enable bit in the Control A register (CTRLA.RUNSTDBY)
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

23.4.2.3 Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

23.4.2.4 Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

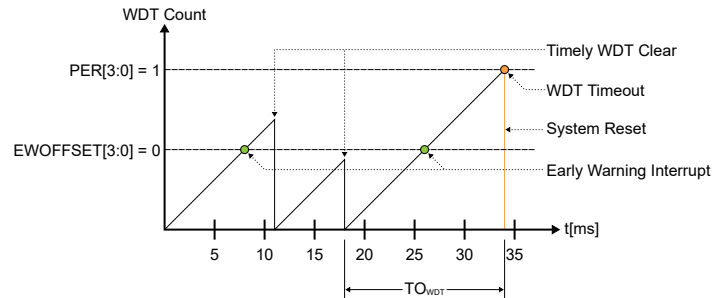
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out (TO_{WDT}) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

Figure 23-2. Normal-Mode Operation



23.4.2.5 Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period (TO_{WDTW}), during the subsequent Normal time-out period (TO_{WDT}). If the WDT is cleared before the time window opens (before TO_{WDTW} is over), the WDT will issue a system reset.

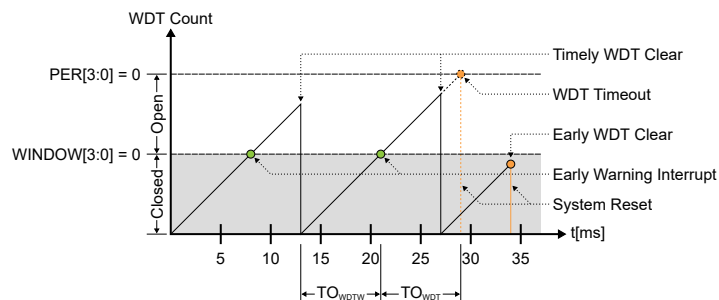
Both parameters TO_{WDTW} and TO_{WDT} are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable SET register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after TO_{WDTW} . The Window mode operation is illustrated in figure Window-Mode Operation.

Figure 23-3. Window-Mode Operation



23.4.3 Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the *Main Clock module (MCLK)*.

A 1024 Hz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

The CLK_WDT_OSC clock is sourced from the clock of the internal Ultra Low-Power Oscillator (OSCULP32K).



CAUTION Watchdog time-out period variations must be considered when implementing software that uses the WDT to ensure that the time-out periods used are valid for all devices. Refer to the *OSCULP32K Electrical Specifications* section of the Electrical Characteristics chapter.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to *Synchronization* section for further details.

Note: The FREQM module can be used to calibrate the output of the OSCULP32K clock against a reference clock, thereby reducing the uncertainty in setting WDT windows.

23.4.4 Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
 - This interrupt is an asynchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the WDT is reset. See the INTFLAG register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together at the system level to generate one combined interrupt request to the *Nested vector Interrupt Controller*. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

23.4.5 Sleep Mode Operation

The Run-In-Standby bit in Control A (CTRLA.RUNSTDBY) control the behavior of the WDT during Standby Sleep mode. When the bit is zero, the watchdog is disabled during sleep, but maintains its current configuration. When CTRLA.RUNSTDBY is '1', the WDT continues to operate during sleep.

23.4.6 Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

23.4.7 Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits") and/or "Write-Synchronized" (or "Write-Synchronized Bits") property in each individual register description.

For more details, refer to *Register Synchronization*.

23.4.8 Additional Features

23.4.8.1 Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are

read-only registers while the CTRLA.ALWAYSON bit is set. Therefore, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

Table 23-2. WDT Operating Modes With Always-On

WEN	Interrupt Enable	Mode
0	0	Always-on and normal mode
0	1	Always-on and normal mode with Early Warning interrupt
1	0	Always-on and window mode
1	1	Always-on and window mode with Early Warning interrupt

23.4.8.2 Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

In Normal mode, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

In window mode, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.

23.5 Peripheral Dependencies

Table 23-3. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
WDT	0x4400E000	11 : EW_A	APB: CLKMSK[0] MASK11	---	11 STATUS0 PERID11			PD_CORE_SW

23.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ALWAYSON	RUNSTDBY				WEN	ENABLE	
0x01	CONFIG	7:0	WINDOW[3:0]				PER[3:0]			
0x02	EWCTRL	7:0	EWOFFSET[3:0]							
0x03	Reserved									
0x04	INTENCLR	7:0								EW
0x05	INTENSET	7:0								EW
0x06	INTFLAG	7:0								EW
0x07	Reserved									
0x08	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0			CLEAR	ALWAYSON	RUNSTDBY	WEN	ENABLE	
0x0C	CLEAR	7:0	CLEAR[7:0]							

23.6.1 Control

Name: CTRLA
Offset: 0x0
Reset: 0x00
Property: RW

Table 23-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ALWAYSON	RUNSTDBY				WEN	ENABLE	
Access	RW	RW				RW	RW	
Reset	0	0				0	0	

Bit 7 – ALWAYSON Always-On

This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a Power-on Reset is received. When this bit is '1', the Control A register (CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.

Writing a '0' to this bit has no effect.

These bit is loaded from User Configuration FUCFG0 at startup.

Note: This bit is not enable-protected.

Value	Description
0	The WDT is enabled and disabled through the ENABLE bit.
1	The WDT is enabled and can only be disabled by a power-on reset (POR).

Bit 6 – RUNSTDBY Run During Standby

This bit controls the behavior of the watchdog during Standby Sleep mode.

- When CTRLA.ALWAYSON = 0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON = 1, this bit is not enable-protected by CTRLA.ENABLE.

These bit is loaded from User Configuration FUCFG0 at start up.

Value	Description
0	The WDT is disabled during Standby sleep mode.
1	The WDT is enabled continues to operate during Standby sleep mode.

Bit 2 – WEN Watchdog Timer Window Mode Enable

This bit enables Window mode.

- When CTRLA.ALWAYSON = 0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON = 1, this bit is not enable-protected by CTRLA.ENABLE.

These bit is loaded from User Configuration FUCFG0 at startup.

Value	Description
0	Window mode is disabled (normal operation).
1	Window mode is enabled.

Bit 1 – ENABLE Enable

This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON = 0.

These bit is loaded from User Configuration FUCFG0 at startup.

Note: This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

Note: This bit is not enable-protected.

Value	Description
0	The WDT is disabled.
1	The WDT is enabled.

23.6.2 Configuration

Name: CONFIG
Offset: 0x1
Reset: 0xBB
Property: RW

Table 23-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	WINDOW[3:0]				PER[3:0]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	0	1	1	1	0	1	1

Bits 7:4 – WINDOW[3:0] Window Mode Time-Out Period

In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1024 Hz CLK_WDT_OSC clock. These bits are loaded from *User Configuration FUCFG0* at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles

Bits 3:0 – PER[3:0] Time-Out Period

These bits determine the watchdog time-out period as a number of 1024 Hz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period. These bits are loaded from *User Configuration FUCFG0* at startup.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles

23.6.3 Early Warning Interrupt Control

Name: EWCTRL
Offset: 0x2
Reset: 0x0B
Property: RW

Table 23-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
					EWOFFSET[3:0]			
Access					RW	RW	RW	RW
Reset					1	0	1	1

Bits 3:0 – EWOFFSET[3:0] Early Warning Interrupt Time Offset

These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from *User Configuration FUCFG0* at start-up.

Value	Name	Description
0x0	CYC8	8 clock cycles
0x1	CYC16	16 clock cycles
0x2	CYC32	32 clock cycles
0x3	CYC64	64 clock cycles
0x4	CYC128	128 clock cycles
0x5	CYC256	256 clock cycles
0x6	CYC512	512 clock cycles
0x7	CYC1024	1024 clock cycles
0x8	CYC2048	2048 clock cycles
0x9	CYC4096	4096 clock cycles
0xA	CYC8192	8192 clock cycles
0xB	CYC16384	16384 clock cycles

23.6.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x4
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Table 23-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								EW
Access								RW
Reset								0

Bit 0 – EW Early Warning Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

23.6.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x5
Reset: 0x00
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Table 23-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								EW
Access								RW
Reset								0

Bit 0 – EW Early Warning Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

Value	Description
0	The Early Warning interrupt is disabled.
1	The Early Warning interrupt is enabled.

23.6.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x6
Reset: 0x00
Property: RW

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 23-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								EW
Access								RW
Reset								0

Bit 0 – EW Early Warning

This flag is cleared by writing a '1' to it.

This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Early Warning interrupt flag.

23.6.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x8
Reset: 0x00000000
Property: R

Table 23-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			CLEAR	ALWAYSON	RUNSTDBY	WEN	ENABLE	
Access			R	R	R	R	R	
Reset			0	0	0	0	0	

Bit 5 – CLEAR Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 4 – ALWAYSON Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 3 – RUNSTDBY Run During Standby Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.RUNSTDBY bit is complete.
1	Write synchronization of the CTRLA.RUNSTDBY bit is ongoing.

Bit 2 – WEN Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.

Value	Description
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Bit 1 - ENABLE Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

23.6.8 Clear

Name: CLEAR
Offset: 0xC
Reset: 0x00
Property: W

Note: This register is write-synchronized: SYNCBUSY.CLEAR must be checked to ensure the CLEAR register synchronization is complete.

Table 23-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CLEAR[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CLEAR[7:0] Watchdog Clear

In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.

In Window mode, any writing attempt to this register before the time-out period started (i.e., during TO_{WDTW}) will issue an immediate system Reset. Writing 0xA5 during the time-out period TO_{WDT} will clear the Watchdog Timer and the complete time-out sequence (first TO_{WDTW} then TO_{WDT}) is restarted.

In both modes, writing any other value than 0xA5 will issue an immediate system Reset.

Note: This bit field is write-synchronized: SYNCBUSY.CLEAR must be checked to ensure the CLEAR.CLEAR synchronization is complete.

Value	Name	Description
0xA5	KEY	Clear Key

24. Frequency Meter (FREQM)

24.1 Overview

The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock source, GCLK_FREQM_MSR, by comparing it to a known reference clock. It counts the number of periods of the measured clock (GCLK_FREQM_MSR) with respect to the reference clock (GCLK_FREQM_REF). The measurement is done for a period of $REFNUM/f_{GCLK\_FREQM\_REF}$ and stored in the Value register (VALUE.VALUE). REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFG.A.REFNUM) over which the duration of a measurement is to be done.

The frequency of the measured clock, f_{CLK_MSR} , is calculated by:

$$f_{GCLK\_FREQM\_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1} \right) f_{GCLK\_FREQM\_REF}$$

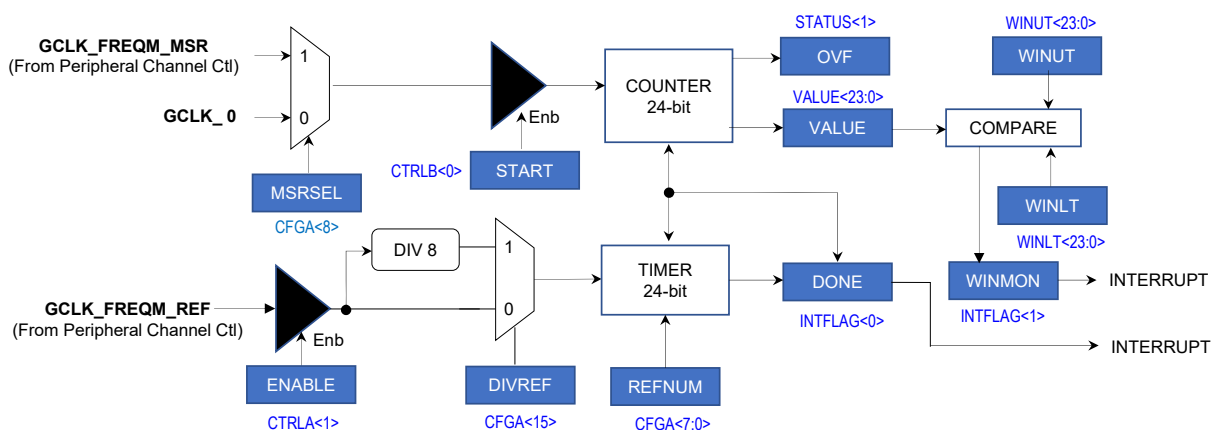
Where *Error* represents the error introduced by the synchronization mechanism. The error can be a maximum of two measured clock cycles.

24.2 Features

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock source from GCLK_FREQM_MSR with respect to a reference clock
- Reference clock can be selected from the available GCLK_FREQM_REF sources
- Measured clock can be selected from the available GCLK_FREQM_MSR sources
- Free-running support enabling continuous clock monitor with flexible window threshold selection

24.3 Block Diagram

Figure 24-1. FREQM Block Diagram



24.4 Signal Description

Signal Name	Type	Description
GCLK_FREQM_MSR	Digital Clock Input	Measured clock source
GCLK_FREQM_REF	Digital Clock Input	Reference measurement clock source from peripheral channel
GCLK0	Digital Clock Input	Internal GCLK0

24.5 Clocks

The clock for the FREQM bus interface (CLK_APB_FREQM) is enabled and disabled by the Main Clock Controller, the default state of CLK_APB_FREQM can be found in [Peripheral Clock Masking](#).

Two generic clocks are used by the FREQM: Reference Clock (GCLK_FREQM_REF) and Measurement Clock (GCLK_FREQM_MSR).

GCLK_FREQM_REF is required to clock the internal reference timer, which acts as the frequency reference.

GCLK_FREQM_MSR is required to clock a ripple counter for frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

The FREQM has also the capability to use other internal clocks as source for frequency measurement. These clocks are chip specific and must be configured and enabled before using the FREQM.

24.6 Functional Description

24.6.1 Basic Operation

24.6.1.1 Initialization

Before enabling FREQM, the device and peripheral must be configured:

- The generic reference clock (GCLK_FREQM_REF) must be configured and enabled.



Important: The reference clock must be slower than the measurement clock.

- Write the number of Reference clock cycles for which the measurement is to be done in the Configuration A register (CFGA.REFNUM).

The following registers are enable-protected, meaning that they can only be written when the FREQM is disabled (CTRLA.ENABLE = 0):

- Configuration A register (CFGA)
- Control C register (CTRLC)
- Event Control register (EVCTRL)
- Window Monitor Lower Threshold register (WINLT)
- Window Monitor Upper Threshold register (WINUT)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the FREQM is disabled (CTRLA.ENABLE = 0):

- On Demand Control bit in Control A register (CTRLA.ONDEMAND)
- Run During Standby bit in Control A register (CTRLA.RUNSTDBY)
- Free Running Mode bit in Control A register (CTRLA.FREERUN)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

24.6.1.2 Enabling, Disabling, and Resetting

The FREQM is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The peripheral is disabled by writing CTRLA.ENABLE = 0.

The FREQM is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). On software reset, all registers in the FREQM will be reset to their initial state, and the FREQM will be disabled.

Then ENABLE and SWRST bits are write-synchronized.

24.6.1.3 Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFG.A.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.

Note:

1. The REFNUM field must be written before the FREQM is enabled.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

Note: Writing a '1' to the START bit in the Control B register during a measurement will be ignored.

The Measurement Done interrupt provides an alternative way to detect measurement completion. When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated. The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

$$f_{CLK\_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1} \right) f_{GCLK\_FREQM\_REF}$$

Where *Error* represents the error introduced by the synchronization mechanism. The error can be a maximum of two CLK_MSR cycles.

Notes:

1. In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) should be checked.
2. Due to asynchronous operation, the VALUE Error measurement can be up to two samples.

In case an overflow condition occurred, indicated by the Overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFG.A.REFNUM), or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then another measurement can be started by writing a '1' to CTRLB.START.

Note: The REFNUM field must be written before the FREQM is enabled.

24.6.1.4 Window Monitor Mode

Window Monitor Mode causes the FREQM to compare each measurement result to pre-configured thresholds. It can compare the value to see if it is below a lower, above an upper, within the upper and lower, or outside of the upper and lower thresholds. The Window Monitor Mode is selected by setting the Window Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT.WINLT) and Window Monitor Upper Threshold register (WINUT.WINUT).

The FREQM also supports an interrupt request for Window Monitor operation: When the Window Monitor bit in Interrupt Enable Set register (INTENSET.WINMON) is '1' and a measurement meets the criteria defined in the Window Mode bits in the Control C register (CTRLC.WINMODE), the Window Monitor bit in the Interrupt Flag Status and Clear register (INTFLAG.WINMON) will be set and an interrupt request is generated.

24.6.1.5 Free Running Mode

Free Running Mode causes the FREQM to automatically start a new measurement after each measurement is complete. The Free Running Mode can be disabled by disabling the module, or by setting the Software Reset bit in the Control A register (CTRLA.SWRST = 1).



Important: In free running mode, when a new measurement starts, the VALUE register will follow the MSR counter value. It is not recommended to read the VALUE register when the free running operating mode is enabled.

Notes:

1. In free running mode, the event detection is disabled. As consequence, all incoming events are ignored and no actions will be triggered inside the FREQM module.
2. Writing to CTRLB.START when a measurement is in progress is ignored. The bit is always read zero.

24.6.2 Interrupts

The FREQM has two interrupt sources:

- DONE: A frequency measurement is done.
- WINMON: A window compare valid condition is detected.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the FREQM is reset. See the Interrupt Flag Status and Clear (INTFLAG) for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the Interrupt Flag Status and Clear (INTFLAG) register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

24.6.3 Events

The FREQM can generate the following output events:

- Done (DONE): Measurement Done.
- Window Monitor (WINMON): Generated when the window monitor condition matches.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The FREQM can take the following actions on an input event:

- **Start measurement (START):** Start a new measurement. If a new START event is detected during an ongoing measurement, the event is ignored.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

By default, the FREQM will detect a rising edge on the incoming event. If the FREQM action must be performed on the falling edge of the incoming event, the event line must be inverted first. This

is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.STARTINV).

24.6.4 Sleep Mode Operation

During Sleep modes, the ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the FREQM clock requests, in cases where the FREQM is enabled (CTRLA.ENABLE = 1) during a sleep mode.

Table 24-1. FREQM Sleep Behavior

CTRLA.RUNSTDBY	CTRLA.ONDEMAND	CTRLA.ENABLE	Description
X	x	0	Disabled
0	0	1	Always request the clocks in all sleep modes, except in STANDBY.
0	1	1	On-demand clocks request in all sleep modes, except STANDBY.
1	0	1	Always request the clocks in all sleep modes.
1	1	1	On-demand clocks request in all sleep modes.

Note: By default, if RUNSTDBY = 0, an ongoing measurement is stopped when the device is going to standby sleep mode. A new measurement will be restarted when the device exits the standby sleep mode.

24.7 Peripheral Dependencies

Table 24-2. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
FREQM	0x4400C000	10 : DONE/ WINMON	APB: CLKMSK[0] MASK10	MSR0: PCHCTRL[2] MSR1: PCHCTRL[3] REF: PCHCTRL[4]	10 STATUS0 PERID10		<u>Generators:</u> DONE: CHANNEL[4] WINMON: CHANNEL[5] <u>Users:</u> START: USER[0],Resynch	PD_CORE_SW

24.8 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	ONDEMAND	RUNSTDBY				FREERUN	ENABLE	SWRST
0x01	CTRLB	7:0								START
0x02	CFGA	15:8	DIVREF					MSRSEL[2:0]		
		7:0	REFNUM[7:0]							
0x04	CTRLC	7:0						WINMODE[2:0]		
0x05	Reserved									
0x06	EVCTRL	7:0			WINMONEO	DONEEO			STARTINV	STARTEI
0x07	Reserved									
0x08	INTENCLR	7:0							WINMON	DONE
0x09	INTENSET	7:0							WINMON	DONE
0x0A	INTFLAG	7:0							WINMON	DONE
0x0B	STATUS	7:0							OVF	BUSY
0x0C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x10	VALUE	31:24								
		23:16	VALUE[23:16]							
		15:8	VALUE[15:8]							
		7:0	VALUE[7:0]							
0x14 ... 0x1F	Reserved									
0x20	WINLT	31:24								
		23:16	WINLT[23:16]							
		15:8	WINLT[15:8]							
		7:0	WINLT[7:0]							
0x24	WINUT	31:24								
		23:16	WINUT[23:16]							
		15:8	WINUT[15:8]							
		7:0	WINUT[7:0]							

24.8.1 Control A Register

Name: CTRLA
Offset: 0x00
Reset: 0x80
Property: RW

Table 24-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY				FREERUN	ENABLE	SWRST
Access	RW	RW				RW	RW	RW
Reset	1	0				0	0	0

Bit 7 – ONDEMAND On Demand Control

During sleep mode, the On Demand mode causes the FREQM clocks to be running when a measurement is requested by a peripheral or if free running operation mode is enabled. If there is no peripheral requesting the FREQM measurement, the module will not request any clock. This bit is enable-protected.

Note: The ONDEMAND bit should always be set to '0'.

Value	Description
0	The FREQM is always requesting the clocks when CTRLA.ENABLE is set.
1	The FREQM is requesting the clocks only when a peripheral is requesting a measurement, or if free running mode is enabled.

Bit 6 – RUNSTDBY Run In Standby

This bit controls how the FREQM clocks will be requested during standby sleep mode. This bit is enable-protected.

Value	Description
0	The FREQM clocks are not requested during standby sleep mode. If CTRLA.ONDEMAND=1, the FREQM will request the clocks only if a measurement is requested. If CTRLA.ONDEMAND=0, the FREQM clocks will be requested as long as CTRLA.ENABLE is set.
1	The FREQM clocks are requested in standby sleep mode. If CTRLA.ONDEMAND=1, the FREQM will request the clocks only if a measurement is requested. If CTRLA.ONDEMAND=0, the FREQM clocks will be requested as long as CTRLA.ENABLE is set.

Bit 2 – FREERUN Free Running Mode

This bit controls the free running operation mode. This bit is enable-protected.

Value	Description
0	The free running operating mode is disabled.
1	The free running operating mode is enabled.

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete. This bit is not enable-protected.

Value	Description
0	The peripheral is disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the FREQM to their initial state, and the FREQM will be disabled. Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not enable-protected.

Value	Description
0	There is no ongoing Reset operation.
1	The Reset operation is ongoing.

24.8.2 Control B Register

Name: CTRLB
Offset: 0x01
Reset: 0x00
Property: W

Table 24-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								START
Access								W
Reset								0

Bit 0 – START Start Measurement

Value	Description
0	Writing a '0' has no effect.
1	Writing a '1' starts a single measurement.

24.8.3 Config A Register

Name: CFGA
Offset: 0x02
Reset: 0x0000
Property: RW

Table 24-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	DIVREF					MSRSEL[2:0]		
Access	RW					RW	RW	RW
Reset	0					0	0	0
Bit	7	6	5	4	3	2	1	0
	REFNUM[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – DIVREF Divide Reference Clock
Divides the reference clock by 8

Value	Name	Description
0x0	DIV1	The reference clock is divided by 1
0x1	DIV8	The reference clock is divided by 8

Bits 10:8 – MSRSEL[2:0] Measurement Clock Selection

Value	Name	Description
0x0	GCLK	GCLK Input Clock
0x1	CPU	CPU Input Clock

Bits 7:0 – REFNUM[7:0] Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles.

24.8.4 Control C Register

Name: CTRLC
Offset: 0x04
Reset: 0x00
Property: RW

Table 24-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						WINMODE[2:0]		
Access								
Reset						0	0	0

Bits 2:0 – WINMODE[2:0] Window Monitor Mode

These bits enable and define the window monitor mode.

Value	Name	Description
0x0	DISABLE	No window mode (default)
0x1	MODE1	VALUE > WINLT
0x2	MODE2	VALUE < WINUT
0x3	MODE3	WINLT < VALUE < WINUT
0x4	MODE4	!(WINLT < VALUE < WINUT)

24.8.5 Event Control Register

Name: EVCTRL
Offset: 0x06
Reset: 0x00
Property: RW

Table 24-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			WINMONEO	DONEEO			STARTINV	STARTEI
Access			RW	RW			RW	RW
Reset			0	0			0	0

Bit 5 – WINMONEO Window Monitor Event Out

Value	Description
0	Window Monitor event output is disabled and an event will not be generated.
1	Window Monitor event output is disabled and an event will be generated.

Bit 4 – DONEEO Measurement Done Event Out

Value	Description
0	Measurement Done event output is disabled and an event will not be generated.
1	Measurement Done event output is disabled and an event will be generated.

Bit 1 – STARTINV Start Measurement Event Invert Enable

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 0 – STARTEI Start Measurement Event Input Enable

Value	Description
0	A new measurement will not be triggered on any incoming event.
1	A new measurement will be triggered on any incoming event.

24.8.6 Interrupt Enable Clear Register

Name: INTENCLR
Offset: 0x08
Reset: 0x00
Property: RW

Table 24-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							WINMON	DONE
Access							RW	RW
Reset							0	0

Bit 1 – WINMON Window Monitor Interrupt Disable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 0 – DONE Measurement Done Interrupt Disable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Measurement Done Interrupt Enable bit, which disables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

24.8.7 Interrupt Enable Set Register

Name: INTENSET
Offset: 0x09
Reset: 0x00
Property: RW

Table 24-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							WINMON	DONE
Access							RW	RW
Reset							0	0

Bit 1 – WINMON Window Monitor Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Window Monitor Interrupt Enable bit, which enables the Window Monitor interrupt.

Value	Description
0	The Window Monitor interrupt is disabled.
1	The Window Monitor interrupt is enabled.

Bit 0 – DONE Measurement Done Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Measurement Done Interrupt Enable bit, which enables the Measurement Done interrupt.

Value	Description
0	The Measurement Done interrupt is disabled.
1	The Measurement Done interrupt is enabled.

24.8.8 Interrupt Flag Register

Name: INTFLAG
Offset: 0x0A
Reset: 0x00
Property: RW

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 24-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							WINMON	DONE
Access							RW	RW
Reset							0	0

Bit 1 – WINMON Window Monitor

This flag is set on the next clock cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the WINMON interrupt flag.

Bit 0 – DONE Measurement Done

This flag is set when the STATUS.BUSY bit has a one-to-zero transition.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the DONE interrupt flag.

24.8.9 Status Register

Name: STATUS
Offset: 0x0B
Reset: 0x00
Property: RW

Table 24-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							OVF	BUSY
Access							RW	R
Reset							0	0

Bit 1 – OVF Sticky Count Value Overflow

This bit is set when an overflow condition occurs to the value counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the OVF status.

Bit 0 – BUSY FREQM Status

Value	Description
0	No ongoing frequency measurement.
1	Frequency measurement is ongoing. This bit is set only when the measurements are triggered by a software start (CTRLB.START = 1).

24.8.10 Synchronization Busy Register

Name: SYNCBUSY
Offset: 0x0C
Reset: 0x00000000
Property: R

Table 24-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable

This bit is cleared when the synchronization of CTRLA.ENABLE is complete.
This bit is set when the synchronization of CTRLA.ENABLE is started.

Bit 0 – SWRST Software Reset

This bit is cleared when the synchronization of CTRLA.SWRST is complete.
This bit is set when the synchronization of CTRLA.SWRST is started.

24.8.11 Count Value Register

Name: VALUE
Offset: 0x10
Reset: 0x00000000
Property: R

Table 24-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	VALUE[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	VALUE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	VALUE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – VALUE[23:0] Measurement Value

Result from measurement.

Note: The register value is refreshed each time a start event is detected, and as long as a measurement is ongoing. It is not recommended to read the register if the start of measurements are triggered by peripheral events or it is in Free Running Mode.

24.8.12 Window Monitor Lower Threshold

Name: WINLT
Offset: 0x20
Reset: 0x00000000
Property: RW

Table 24-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	WINLT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WINLT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	WINLT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – WINLT[23:0] Window Lower Threshold

When the Window Monitor bit in Interrupt Enable Set register, INTENSET.WINMON= 1, and the VALUE register result of the measurement depends on CTRLC.WINMON, the Window Monitor bit in the Interrupt Flag Status and Clear register, INTFLAG.WINMON, will be set and an interrupt request is generated.

24.8.13 Window Monitor Upper Threshold

Name: WINUT
Offset: 0x24
Reset: 0x00000000
Property: RW

Table 24-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	WINUT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WINUT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	WINUT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – WINUT[23:0] Window Upper Threshold

When the Window Monitor bit in Interrupt Enable Set register, INTENSET.WINMON= 1, and the VALUE register result of the measurement depends on CTRL.C.WINMON, the Window Monitor bit in the Interrupt Flag Status and Clear register, INTFLAG.WINMON, will be set and an interrupt request is generated.

25. Real-Time Counter (RTC)

25.1 Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, overflow wake up mechanisms, or from the wake inputs.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and peripheral event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768 kHz clock source, the minimum counter tick interval is 30.5 μ s and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

25.2 Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- Two 32-bit or four 16-bit compare values
- Clock/Calendar mode
 - Time in seconds, minutes, and hours (12/24)
 - Date in day of month, month, and year
 - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
 - Optional clear on alarm/compare match
- Four general purpose registers
- Tamper Detection
 - Up to 8 static tamper inputs with programmable level detection
 - Up to 8 inputs and 8 outputs dynamic tamper (Active layer protection)
 - Support for any kinds of static and dynamic combinations
 - Timestamp on tamper events

25.3 Block Diagram

Figure 25-1. RTC Block Diagram (Mode 0-32 Bit Counter)

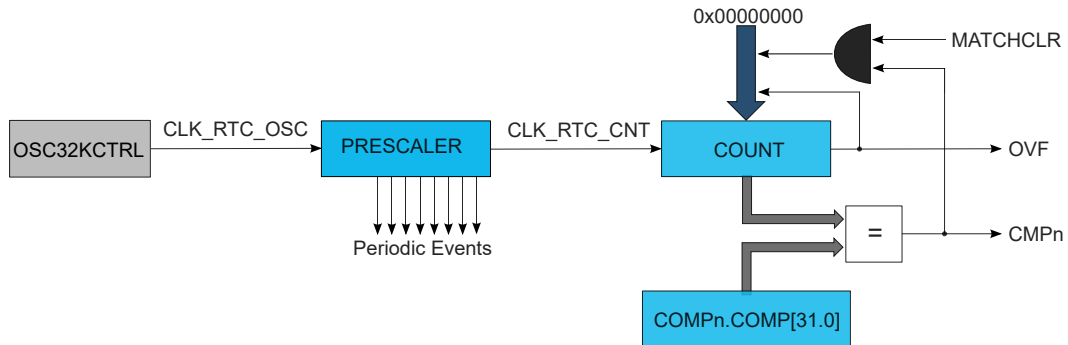


Figure 25-2. RTC Block Diagram (Mode 1-16 Bit Counter)

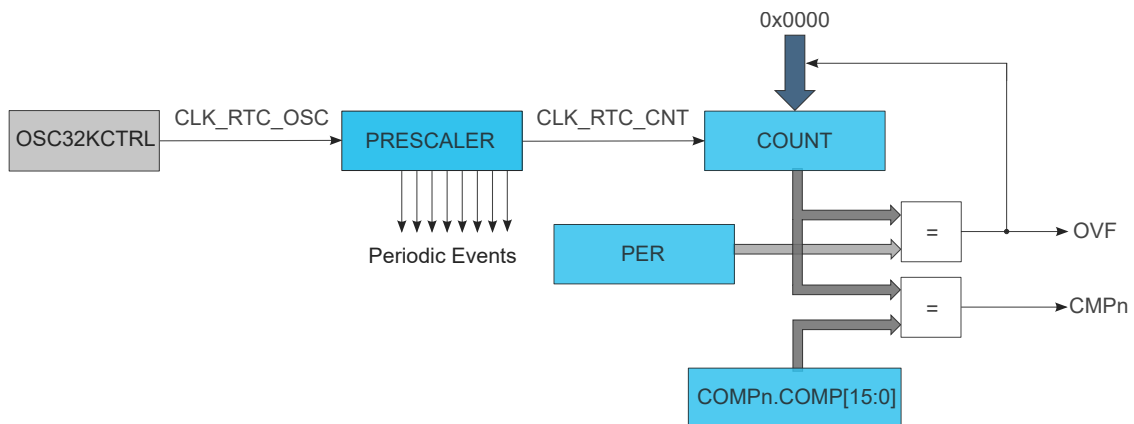


Figure 25-3. RTC Block Diagram (Mode 2 — Clock/Calendar)

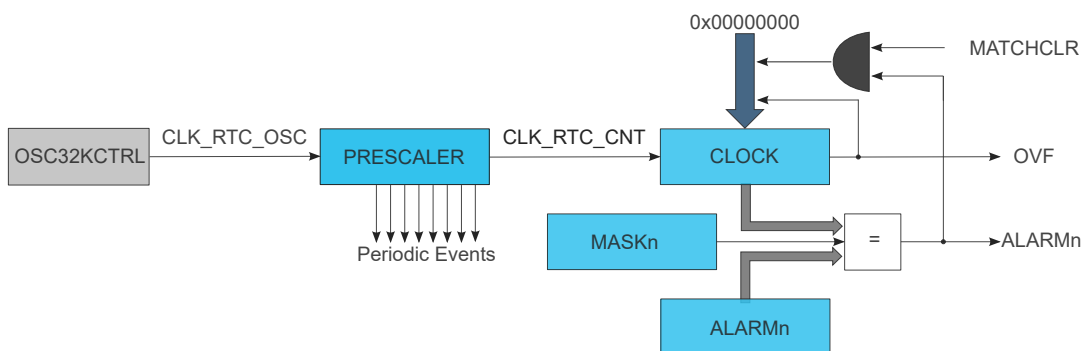
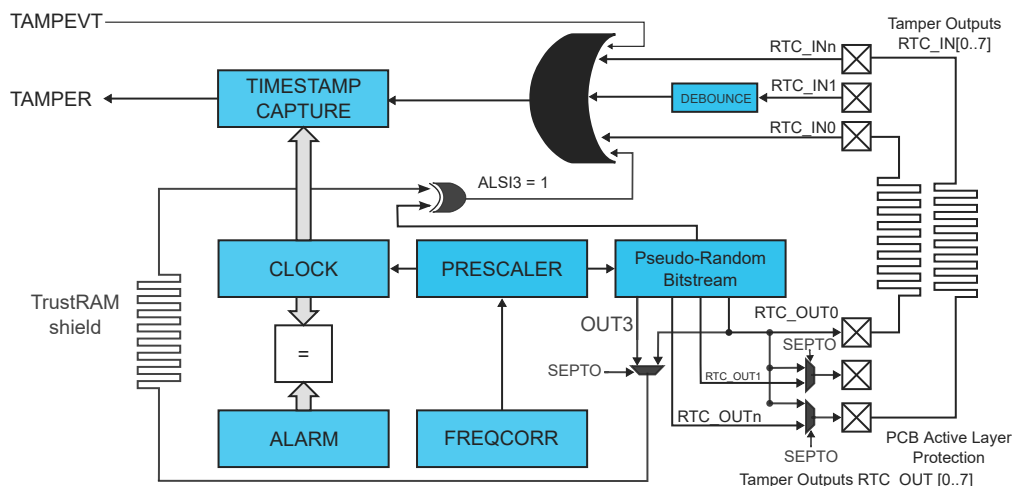


Figure 25-4. RTC Block Diagram (Tamper Detection Use Case)



25.4 Signal Description

Table 25-1. Signal Description

Signal	Description	Type
RTC_IN[7:0] ⁽¹⁾	Tamper Detection Input	Digital input
RTC_OUT[7:0] ⁽¹⁾	Tamper Detection Output	Digital output

25.5 Functional Description

25.5.1 Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

25.5.2 Basic Operation

25.5.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE = 0):

- The Operating Mode bits in the Control A register (CTRLA.MODE)
- The Prescaler bits in the Control A register (CTRLA.PRESCALER)
- The Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- The Clock Representation bit in the Control A register (CTRLA.CLKREP)
- The GP Registers Reset On Tamper Enable bit in the Control A register (CTRLA.GPTRST)

The following registers are enable-protected:

- The Control B (CTRLB) register
- The Event Control (EVCTRL) register

- The Tamper Control (TAMPCTRL) register
- The Tamper Control B (TAMPCTRLB) register

The Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE = 0). If the RTC is enabled (CTRLA.ENABLE = 1), these operations are necessary: first write CTRLA.ENABLE = 0 and check whether the write synchronization has finished, then change the desired bit field value. The Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

Note: In Clock/Calendar mode, the prescaler must be configured to provide a 1 Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$$f_{\text{CLK\_RTC\_CNT}} = \frac{f_{\text{CLK\_RTC\_OSC}}}{2^{\text{PRESCALER}-1}} \text{ when PRESCALER is not 0, else } f_{\text{CLK\_RTC\_CNT}} = f_{\text{CLK\_RTC\_OSC}}.$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{\text{CLK\_RTC\_OSC}}$, and $f_{\text{CLK\_RTC\_CNT}}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

25.5.2.2 Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE = 1). The RTC is disabled by writing CTRLA.ENABLE = 0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST = 1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

25.5.2.3 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in the Block Diagram. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare registers (COMPn, n=0..1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMPn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMPn and INTFLAG.OVF will both be set simultaneously on a compare match with COMPn.

25.5.2.4 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x1, the counter operates in 16-bit Counter mode as shown in the Block Diagram. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..3). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..3) is set on the next 0-to-1 transition of CLK_RTC_CNT.

25.5.2.5 Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode, as shown in the Block Diagram. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds
- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm registers (ALARMn, n=0,1). When an alarm match occurs, the Alarm n Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARMn) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm n Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm n Mask register (MASKn.SEL, n=0,1). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARMn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see Periodic Intervals).

Note: When CTRLA.MATCHCLR is 1, INTFLAG.ALARMn and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARMn.

25.5.3 DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

If the CPU accesses the registers which are source for DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

25.5.4 Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Indicates detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to *Periodic Intervals* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one or more combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for details.

25.5.5 Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Generated on detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to *Periodic Intervals* for details.
- Periodic Daily (PERD): Generated when the COUNT/CLOCK has incremented at a fixed period of time.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the EVSYS - Event System for details on configuring the event system.

The RTC can take the following actions on an input event:

- Tamper (TAMPEVT): Capture the RTC counter to the timestamp register. See *Tamper Detection*.

Writing a one to an Event Input Enable bit ((EVCTRL.TAMPEVEI)) into the Event Control register (EVCTRL) enables the action on input events. Writing a zero to this bit disables the action on input events.

25.5.6 Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue executing right from the first instruction that followed the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *Event System* for more information.

25.5.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE
- Count Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)
- Clock Read Synchronization bit in Control A register (CTRLA.CLOCKSYNC)

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn
- The General Purpose n registers (GPn)

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

25.5.8 Additional Features

25.5.8.1 Periodic Intervals

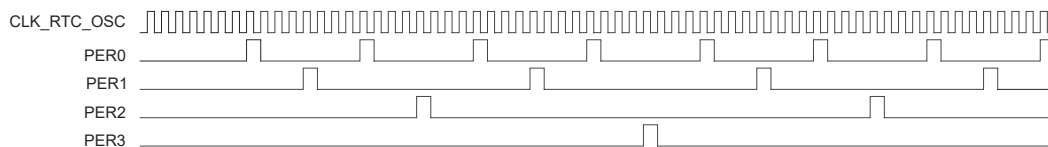
The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

$$f_{\text{PERIODIC}(n)} = \frac{f_{\text{CLK\_RTC\_OSC}}}{2^{n+3}}$$

$f_{\text{CLK\_RTC\_OSC}}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

Figure 25-5. Example Periodic Events



25.5.8.2 Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 8192 CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over 128 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{8192 \cdot 128} \cdot 10^6 \text{ ppm}$$

This results in a resolution of 0.95367ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

25.5.8.3 General Purpose Registers

The RTC includes four General Purpose registers (GPn). These registers are reset only when the RTC is reset or when tamper detection occurs while CTRLA.GPTRST=1, and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

The general purpose registers 2*n and 2*n+1 are enabled by writing a '1' to the General Purpose Enable bit n in the Control B register (CTRLB.GPnEN).

The GP registers share internal resources with the COMPARE/ALARM features. Each COMPARE/ALARM register have a separate read buffer and write buffer. When the general purpose feature is enabled the even GP uses the read buffer while the odd GP uses the write buffer.

When the COMPARE/ALARM register is written, the write buffer hold temporarily the COMPARE/ALARM value until the synchronization is complete (bit SYNCBUSY.COMPn going to 0). After the write is completed the write buffer can be used as a odd general purpose register without affecting the COMPARE/ALARM function.

If the COMPARE/ALARM function is not used, the read buffer can be used as an even general purpose register. In this case writing the even GP will temporarily use the write buffer until the synchronization is complete (bit SYNCBUSY.GPn going to 0). Thus an even GP must be written before writing the odd GP. Changing or writing an even GP needs to temporarily save the value of the odd GP.

Before using an even GP, the associated COMPARE/ALARM feature must be disabled by writing a '1' to the General Purpose Enable bit in the Control B register (CTRLB.GPnEN). To re-enable the compare/alarm, CTRLB.GPnEN must be written to zero and the associated COMPn/ALARMn must be written with the correct value.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to COMP0 to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0 if needed.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

The following table provides the correspondence of General Purpose Registers and the COMPARE/ALARM read or write buffer in all RTC modes.

Table 25-2. General Purpose Registers Versus Compare/Alarm Registers: n in 0, 2, 4, 6...

Register	Mode 0	Mode 1	Mode 2	Write Before
GPn	COMPn/2 write buffer	(COMPn , COMPn+1) write buffer	ALARM0 write buffer	GPn+1
GPn+1	COMPn/2 read buffer	(COMPn , COMPn+1) read buffer	ALARM0 read buffer	-

25.5.8.4 Tamper Detection

The RTC provides eight tamper channels that can be used for tamper detection.

The action of each tamper channel is configured using the Input n Action bits in the Tamper Control register (TAMPCTRL.INnACT):

- **Off:** Detection for tamper channel n is disabled
- **Wake:** A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will not be captured in the TIMESTAMP register.
- **Capture:** A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.
- **Active Layer Protection:** A mismatch of an internal RTC signal routed between INn and OUTn pins will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The connection between IN and OUT can be made externally over the PCB, or internally over the TrustRAM. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each tamper channel. These bits remain active until cleared by software.

A single interrupt request (TAMPER) is available for all tamper channels.

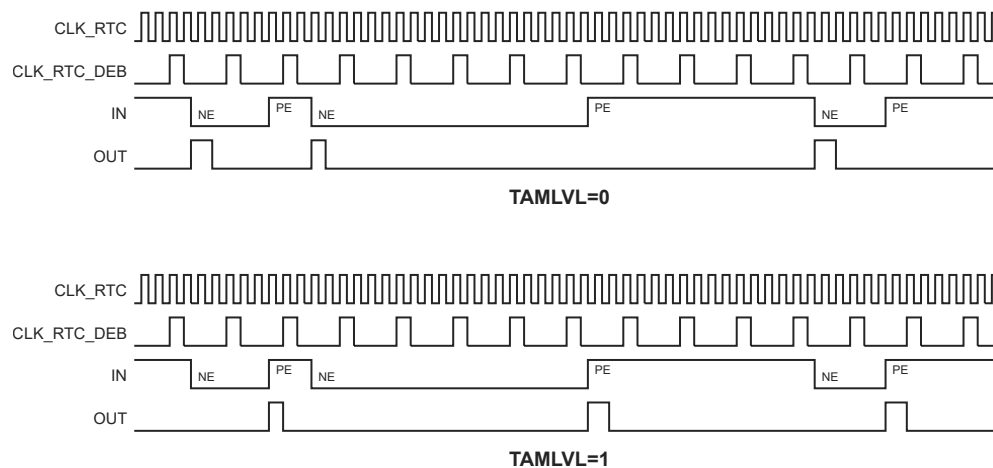
The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVTEI).

Up to eight external inputs (INn) can be used for tamper detection. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn).

Separate debouncers are embedded for each external input. The debouncer for each input is enabled/disabled with the Debounce Enable bits in the Tamper Control register (TAMPCTRL.DEBNCn). The debouncer configuration is fixed for all inputs as set by the Control B register (CTRLB). The debouncing period duration is configurable using the Debounce Frequency field in the Control B register (CTRLB.DEBF). The period is set for all debouncers (i.e., the duration cannot be adjusted separately for each debouncer).

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. See the following figure for an example.

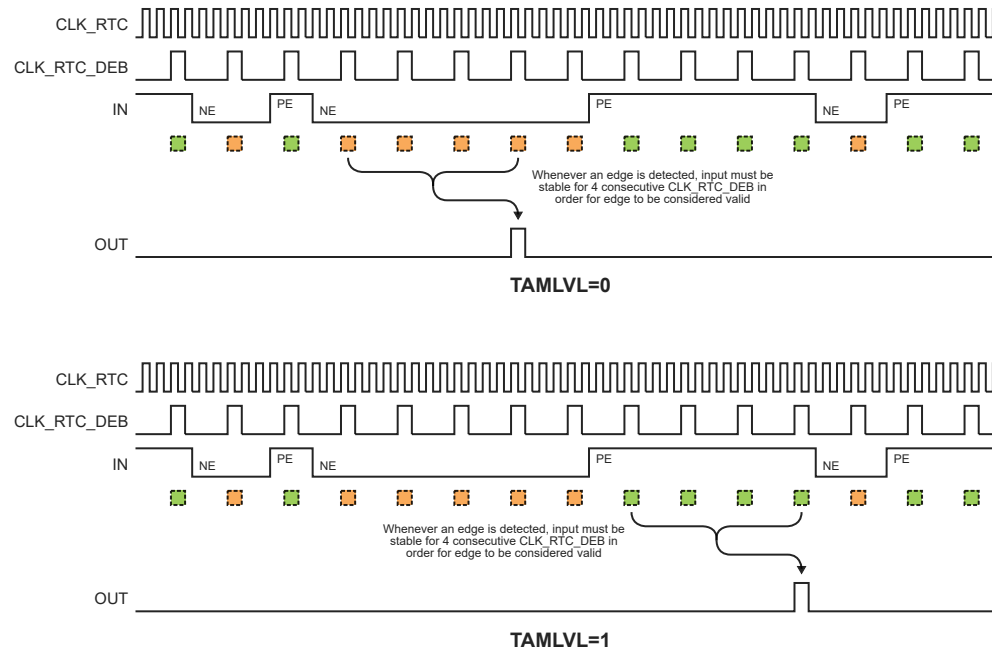
Figure 25-6. Edge Detection with Debouncer Disabled



When TAMPCTRL.DEBNCn = 1, the detection time depends on whether the debouncer operates synchronously or asynchronously, and whether majority detection is enabled or not. Refer to the table below for more details. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

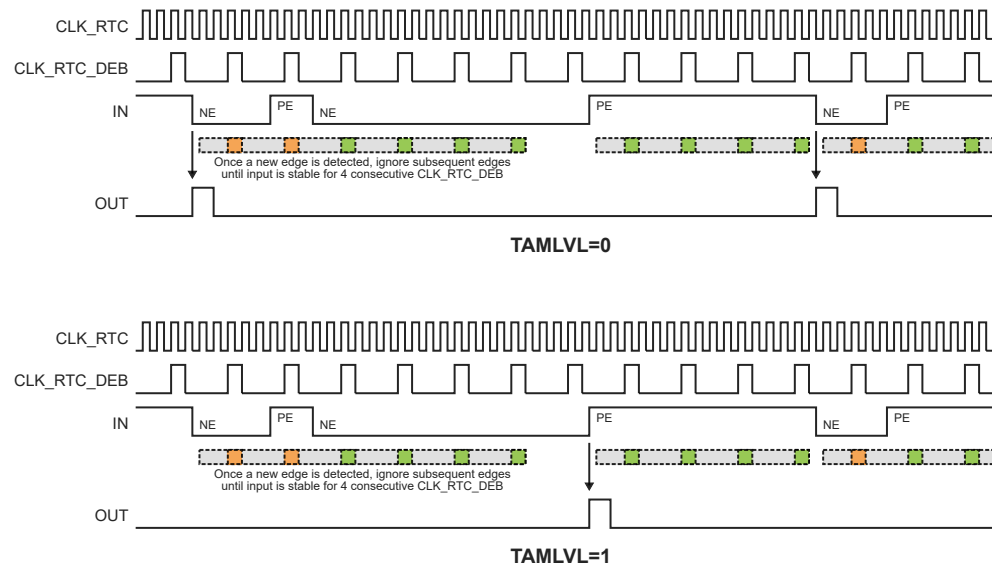
- **Synchronous (CTRLB.DEBASYNC = 0):** INn is synchronized in two CLK_RTC periods and then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. See the following figure for an example.

Figure 25-7. Edge Detection with Synchronous Stability Debouncing



- **Asynchronous (CTRLB.DEBASYNC = 1):** The first edge on INn is detected. Further detection is blanked until INn remains stable for four CLK_RTC_DEB periods. See the following figure for an example.

Figure 25-8. Edge Detection with Asynchronous Stability Debouncing



Majority debouncing is configured by the Debounce Majority Enable bit in the Control B register (CTRLB.DEBMAJ). INn must be valid for two out of three CLK_RTC_DEB periods. See the following figure for an example.

Figure 25-9. Edge Detection with Majority Debouncing

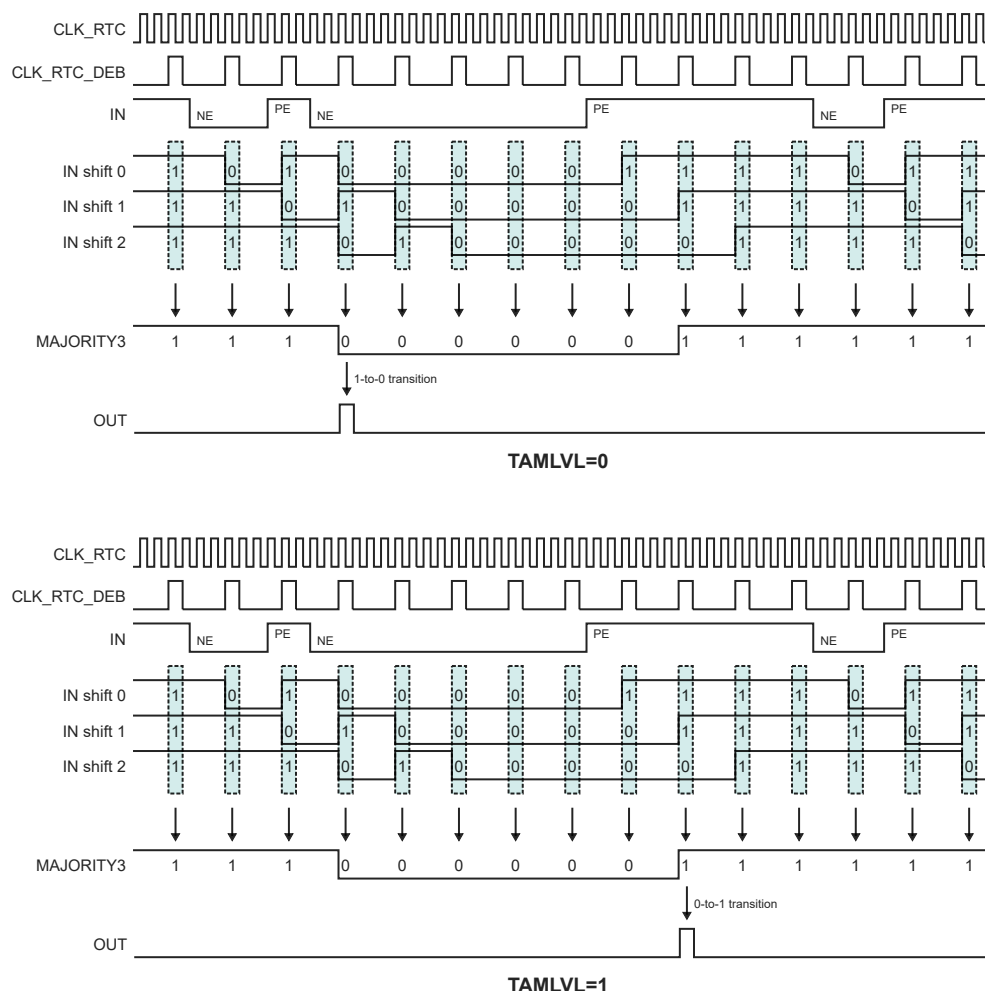


Table 25-3. Debouncer Configuration

TAMPCTRL. DEBNCn	CTRLB. DEBMAJ	CTRLB. DEBASYN	Description
0	X	X	Detect edge on INn with no debouncing. Every edge detected is immediately triggered.
1	0	0	Detect edge on INn with synchronous stability debouncing. Edge detected is only triggered when INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	0	1	Detect edge on INn with asynchronous stability debouncing. First detected edge is triggered immediately. All subsequent detected edges are ignored until INn is stable for 4 consecutive CLK_RTC_DEB periods.
1	1	X	Detect edge on INn with majority debouncing. Pin INn is sampled for 3 consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = '0' and LHH, HLH, HHL, HHH = '1').

25.5.8.4.1 Timestamp

As part of tamper detection the RTC can capture the counter value (COUNT/CLOCK) into the TIMESTAMP register. Three CLK_RTC periods are required to detect the tampering condition and capture the value. The TIMESTAMP value can be read once the Tamper flag in the Interrupt Flag

register (INTFLAG.TAMPER) is set. If the DMA Enable bit in the Control B register (CTRLB.DMAEN) is '1', a DMA request will be triggered by the timestamp. In order to determine which tamper source caused a capture, the Tamper ID register (TAMPID) provides the detection status of each tamper channel and the tamper input event. A DMA transfer can then read both TIMESTAMP and TAMPID in succession.

A new timestamp value cannot be captured until the Tamper flag is cleared, either by reading the timestamp or by writing a '1' to INTFLAG.TAMPER. If several tamper conditions occur in a short window before the flag is cleared, only the first timestamp may be logged. However, the detection of each tamper will still be recorded in TAMPID.

The Tamper Input Event (TAMPEVT) will always perform a timestamp capture. To capture on the external inputs (INn), the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT) must be written to 'CAPTURE' or 'ACTL'. If an input is set for wake functionality it does not capture the timestamp; however the Tamper flag and TAMPID will still be updated.

Note: The TIMESTAMP value should be read once, and INTFLAG.TAMPER must be cleared. The next value should be read only after the INTFLAG.TAMPER is set again.

25.5.8.4.2 Active Layer Protection

The RTC provides a mean of detecting broken traces on the PCB, also known as Active layer Protection. In this mode, a generated internal RTC signal can be directly routed over critical components on the board using RTC OUT output pin to one RTC INn input pin. A tamper condition is detected if there is a mismatch on the generated RTC signal.

The Active Layer Protection mode and the generation of the RTC signal is enabled by setting the RTCOUT bit in the Control B register (CTRLB.RTCOUT).

The selection of the TrustRAM active layer protection is done by setting the Active Layer Internal Select n bit in the TAMPCTRLB register (TAMPCTRLB.ALSIn), while clearing this bit selects the RTC INn and OUTn external protection.



Important: Only one tamper channel must be configured on the TrustRAM, while remaining channels can be independently configured on tamper detection pins.

Note: The Active Layer Protection works with distinct OUTn/INn pins when the Separate Tamper Output bit CTRLB.SEPTO is set. Alternatively, it can work with one output pin (OUT0) and multiple input pin INn. This is achieved by clearing the Separate Tamper Output bit CTRLB.SEPTO. If the TrustRAM active layer protection is selected for a channel and CTRLB.SEPTO is cleared, it will use the same internal signal as OUT0, otherwise it uses the OUTn signal.

Enabling active layer protection requires the following steps:

- Enable the RTC prescaler output by writing a one to the RTC Out bit in the Control B register (CTRLB.RTCOUT). The I/O pins must also be configured to correctly route the signal to the external pins.
- Select the frequency of the output signal by configuring the RTC Active Layer Frequency field in the Control B register (CTRLB.ACTF).

$$GCLK\_RTC\_OUT = \frac{CLK\_RTC}{2^{CTRLB.ACTF + 1}}$$

- Set the Separate Tamper Output bit CTRLB.SEPTO in case of separate Active Layer Protection channels
- Clear the Active Layer Internal Select bit n in the TAMPCTRLB register (TAMPCTRLB.ALSIn) to enable the protection using INn and OUTn pins. Set the Active Layer Internal Select bit n in the TAMPCTRLB register (TAMPCTRLB.ALSIn) to enable the protection of the TrustRAM shield.

- Enable the tamper input n (INn) in active layer mode by writing 3 to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT). When active layer protection is enabled and the INn and OUTn pin are used, the value of the INn is sampled on the rising edge of CLK_RTC and compared to the expected value of OUTn. Therefore up to one CLK_RTC period is available for propagation delay through the trace.
- Enable the RTC prescaler output and the Active Layer Protection by setting the CTRLB.RTCOUT bit. The I/O pins must also be configured to correctly route the signal to the external pins.

25.6 Peripheral Dependencies

Table 25-4. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
RTC	0x45006000	12 : TAMPER 13 : OVF 14 : PER0/ PER1/PER2/ PER3/PER4/ PER5/PER6/ PER7 15 : CMP0/ CMP1/CMP2/ CMP3	APB: CLKMSK[0] MASK12	---	12 STATUS0 PERID12	3 : TIMESTAMP	<u>Generators:</u> PER0: CHANNEL[6] PER1: CHANNEL[7] PER2: CHANNEL[8] PER3: CHANNEL[9] PER4: CHANNEL[10] PER5: CHANNEL[11] PER6: CHANNEL[12] PER7: CHANNEL[13] CMP0: CHANNEL[14] CMP1: CHANNEL[15] CMP2: CHANNEL[16] CMP3: CHANNEL[17] TAMPER: CHANNEL[18] OVF: CHANNEL[19] PERD: CHANNEL[20] <u>Users:</u> TAMPER: USER[1],Async	PD_CORE_BU

25.7 Register Summary - MODE0

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0	
0x00	CTRLA	15:8	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]				
		7:0	MATCHCLR				MODE[1:0]		ENABLE	SWRST	
0x02	CTRLB	15:8	SEPTO	ACTF[2:0]				DEBF[2:0]			
		7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ	GP6EN	GP4EN	GP2EN	GP0EN	
0x04	EVCTRL	31:24								PERDEO	
		23:16								TAMPEVEI	
		15:8	OVFEO	TAMPEREO	CMPEO[3:0]						
		7:0	PEREO[7:0]								
0x08	INTENCLR	15:8	OVF	TAMPER				CMP[3:0]			
		7:0	PER[7:0]								
0x0A	INTENSET	15:8	OVF	TAMPER				CMP[3:0]			
		7:0	PER[7:0]								
0x0C	INTFLAG	15:8	OVF	TAMPER				CMP[3:0]			
		7:0	PER[7:0]								
0x0E	Reserved										
...											
0x0F											
0x10	SYNDBUSY	31:24									
		23:16	GP[7:0]								
		15:8	COUNTSYNC							COMP[3]	
		7:0	COMP[2:0]				COUNT	FREQCORR	ENABLE	SWRST	
0x14	Reserved										
...											
0x17											
0x18	COUNT	31:24	COUNT[31:24]								
		23:16	COUNT[23:16]								
		15:8	COUNT[15:8]								
		7:0	COUNT[7:0]								
0x1C	Reserved										
...											
0x1F											
0x20	COMP	31:24	COMP[31:24]								
		23:16	COMP[23:16]								
		15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x24	Reserved										
...											
0x63											
0x64	TIMESTAMP	31:24	COUNT[31:24]								
		23:16	COUNT[23:16]								
		15:8	COUNT[15:8]								
		7:0	COUNT[7:0]								

25.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: RW

Table 25-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST	BKTRST			PRESCALER[3:0]		
Access	RW	RW	RW		RW	RW	RW	RW
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR				MODE[1:0]		ENABLE	SWRST
Access	RW				RW	RW	RW	W
Reset	0				0	0	0	0

Bit 15 – COUNTSYNC Count Read Synchronization Enable

The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable

Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

Bit 13 – BKTRST BKUP Registers Reset On Tamper Enable

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128

Value	Name	Description
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024

Bit 7 – MATCHCLR Clear on Match

This bit defines if the counter is cleared or not on a match.
This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bits 3:2 – MODE[1:0] Operating Mode

This bit group defines the operating mode of the RTC.
This bit is not synchronized.

Value	Name	Description
0x0	COUNT32	Mode 0: 32-bit Counter
0x1	COUNT16	Mode 1: 16-bit Counter
0x2	CLOCK	Mode 2: Clock/Calendar

Bit 1 – ENABLE Enable

Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization The Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user must poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.7.2 Control B

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: RW

Table 25-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[2:0]		
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ	GP6EN	GP4EN	GP2EN	GP0EN
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – SEPTO Separate Tamper Outputs

Bits 14:12 – ACTF[2:0] Active Layer Frequency

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC/2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC/4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC/8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC/16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC/32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC/64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC/128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC/256

Bits 10:8 – DEBF[2:0] Debounce Frequency

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC/2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC/4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC/8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC/16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC/32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC/64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC/128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC/256

Bit 7 – DMAEN DMA Enable

Bit 6 – RTCOUT RTC Output Enable

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Bit 4 – DEBMAJ Debouncer Majority Enable

Bit 3 – GP6EN General Purpose 6 Enable

Bit 2 – GP4EN General Purpose 4 Enable

Bit 1 – GP2EN General Purpose 2 Enable

Bit 0 – GP0EN General Purpose 0 Enable

25.7.3 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 25-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								RW
Reset								0

Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								RW
Reset								0

Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO			CMPEO[3:0]			
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0

Bit	7	6	5	4	3	2	1	0
	PEREO[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000).

Bit 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled and incoming events will be ignored.
1	Tamper event input is enabled and incoming events will capture the COUNT value.

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled and will not be generated.
1	Tamper event output is enabled and will be generated for every tamper input.

Bits 11:8 – CMPEO[3:0] Compare x Event Output Enable

Bits 7:0 – PEREO[7:0] Periodic Interval x Event Output Enable

25.7.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Table 25-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER				CMP[3:0]		
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 11:8 – CMP[3:0] Compare x Interrupt Enable

Bits 7:0 – PER[7:0] Periodic Interval x Interrupt Enable

25.7.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Table 25-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER				CMP[3:0]		
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 11:8 – CMP[3:0] Compare x Interrupt Enable

Bits 7:0 – PER[7:0] Periodic Interval x Interrupt Enable

25.7.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 25-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			CMP[3:0]			
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow

This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper

This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bits 11:8 – CMP[3:0] Compare x

Bits 7:0 – PER[7:0] Periodic Interval x

25.7.7 Synchronization Busy Status

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: R

Table 25-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	GP[7:0]							
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	COUNTSYNC							COMP[3]
Reset	R							R
Reset	0							0
Bit	7	6	5	4	3	2	1	0
Access	COMP[2:0]				COUNT	FREQCORR	ENABLE	SWRST
Reset	R	R	R		R	R	R	R
Reset	0	0	0		0	0	0	0

Bits 23:16 – GP[7:0] General Purpose x Register Busy

Bit 15 – COUNTSYNC Count Synchronization Enable Bit Busy

Value	Description
0	Write synchronization for the CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for the CTRLA.COUNTSYNC bit is ongoing.

Bits 8:5 – COMP[3:0] COMP x Register Busy

Bit 3 – COUNT COUNT Register Busy

Value	Description
0	Read/write synchronization for the COUNT register is complete.
1	Read/write synchronization for the COUNT register is ongoing.

Bit 2 – FREQCORR FREQCORR Register Busy

Value	Description
0	Write synchronization for the FREQCORR register is complete.
1	Write synchronization for the FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Bit Busy

Value	Description
0	Write synchronization for the CTRLA.ENABLE bit is complete.
1	Write synchronization for the CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Busy

Value	Description
0	Write synchronization for the CTRLA.SWRST bit is complete.
1	Write synchronization for the CTRLA.SWRST bit is ongoing.

25.7.8 Counter Value

Name: COUNT
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 25-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value

These bits define the value of the 32-bit RTC counter in mode 0.

25.7.9 Compare n Value

Name: COMP
Offset: 0x20
Reset: 0x00000000
Property: RW

Table 25-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	COMP[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COMP[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COMP[31:0] Compare Value

The 32-bit value of COMP0 is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is '1'.

25.7.10 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x00000000
Property: R

Table 25-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Count Timestamp Value

The 32-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs

25.8 Register Summary - MODE1

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0	
0x00	CTRLA	15:8	COUNTSYNC	GPTRST	BKTRST		PRESCALER[3:0]				
		7:0					MODE[1:0]		ENABLE	SWRST	
0x02	CTRLB	15:8	SEPTO	ACTF[2:0]				DEBF[2:0]			
		7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ	GP6EN	GP4EN	GP2EN	GP0EN	
0x04	EVCTRL	31:24								PERDEO	
		23:16								TAMPEVEI	
		15:8	OVFEO	TAMPEREO	CMPEO[5:0]						
		7:0	PEREO[7:0]								
0x08	INTENCLR	15:8	OVF	TAMPER	CMP[5:0]						
		7:0	PER[7:0]								
0x0A	INTENSET	15:8	OVF	TAMPER	CMP[5:0]						
		7:0	PER[7:0]								
0x0C	INTFLAG	15:8	OVF	TAMPER	CMP[5:0]						
		7:0	PER[7:0]								
0x0E ... 0x0F	Reserved										
0x10	SYNCBUSY	31:24									
		23:16	GP[7:0]								
		15:8	COUNTSYNC					COMP[5:3]			
		7:0	COMP[2:0]			PER	COUNT	FREQCORR	ENABLE	SWRST	
0x14 ... 0x17	Reserved										
0x18	COUNT	15:8	COUNT[15:8]								
		7:0	COUNT[7:0]								
0x1A ... 0x1B	Reserved										
0x1C	PER	15:8	PER[15:8]								
		7:0	PER[7:0]								
0x1E ... 0x1F	Reserved										
0x20	COMP0	15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x22	COMP1	15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x24	COMP2	15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x26	COMP3	15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x28	COMP4	15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x2A	COMP5	15:8	COMP[15:8]								
		7:0	COMP[7:0]								
0x2C ... 0x63	Reserved										
0x64	TIMESTAMP	31:24									
		23:16									
		15:8	COUNT[15:8]								
		7:0	COUNT[7:0]								

25.8.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: RW

Table 25-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	COUNTSYNC	GPTRST	BKTRST			PRESCALER[3:0]		
Access	RW	RW	RW		RW	RW	RW	RW
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
					MODE[1:0]		ENABLE	SWRST
Access					RW	RW	RW	W
Reset					0	0	0	0

Bit 15 – COUNTSYNC Count Read Synchronization Enable

The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register. This bit is not enable-protected.

Value	Description
0	COUNT read synchronization is disabled
1	COUNT read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable

Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	GPn registers will not reset when a tamper condition occurs.
1	GPn registers will reset when a tamper condition occurs.

Bit 13 – BKTRST BKUP Registers Reset On Tamper Enable

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off. These bits are not synchronized.

Value	Name	Description
0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024
1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
3	DIV4	CLK_RTC_CNT = GCLK_RTC/4

Value	Name	Description
4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
8	DIV128	CLK_RTC_CNT = GCLK_RTC/128
9	DIV256	CLK_RTC_CNT = GCLK_RTC/256

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0	COUNT32	Mode 0: 32-bit Counter
1	COUNT16	Mode 1: 16-bit Counter
2	CLOCK	Mode 2: Clock/Calendar

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete.

CTRLA.SWRST will be cleared when the reset is complete.

This bit is not enable-protected.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.8.2 Control B

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: RW

Table 25-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[2:0]		
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASYNC	DEBMAJ	GP6EN	GP4EN	GP2EN	GP0EN
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – SEPTO Separate Tamper Outputs

Bits 14:12 – ACTF[2:0] Active Layer Frequency

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC/2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC/4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC/8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC/16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC/32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC/64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC/128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC/256

Bits 10:8 – DEBF[2:0] Debounce Frequency

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC/2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC/4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC/8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC/16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC/32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC/64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC/128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC/256

Bit 7 – DMAEN DMA Enable

Bit 6 – RTCOUT RTC Output Enable

Bit 5 – DEBASYNC Debouncer Asynchronous Enable

Bit 4 – DEBMAJ Debouncer Majority Enable

Bit 3 – GP6EN General Purpose 6 Enable

Bit 2 – GP4EN General Purpose 4 Enable

Bit 1 – GP2EN General Purpose 2 Enable

Bit 0 – GP0EN General Purpose 0 Enable

25.8.3 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 25-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								RW
Reset								0
Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								RW
Reset								0
Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO	CMPEO[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PEREO[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000).

Bit 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored
1	Tamper event input is enabled, and incoming events will capture the COUNT value

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated.
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 13:8 – CMPEO[5:0] Compare x Event Output Enable

Bits 7:0 – PEREO[7:0] Periodic Interval x Event Output Enable

25.8.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Table 25-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER	CMP[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 13:8 – CMP[5:0] Compare x Interrupt Enable

Bits 7:0 – PER[7:0] Periodic Interval x Interrupt Enable

25.8.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Table 25-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER	CMP[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 13:8 – CMP[5:0] Compare x Interrupt Enable

Bits 7:0 – PER[7:0] Periodic Interval x Interrupt Enable

25.8.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: RW

Table 25-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER	CMP[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow

This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper

This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/ INTENSET.TAMPER is one.

Writing a '0' to this bit has no effect.

Writing a one to this bit clears the Tamper interrupt flag.

Bits 13:8 – CMP[5:0] Compare x

Bits 7:0 – PER[7:0] Periodic Interval x

25.8.7 Synchronization Busy Status

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: R

Table 25-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	GP[7:0]							
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	COUNTSYNC					COMP[5:3]		
Reset	R					R	R	R
Reset	0					0	0	0
Bit	7	6	5	4	3	2	1	0
Access	COMP[2:0]			PER	COUNT	FREQCORR	ENABLE	SWRST
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – GP[7:0] General Purpose x Register Busy

Bit 15 – COUNTSYNC Count Synchronization Enable Bit Busy

Value	Description
0	Write synchronization for CTRLA.COUNTSYNC bit is complete.
1	Write synchronization for CTRLA.COUNTSYNC bit is ongoing.

Bits 10:5 – COMP[5:0] COMP x Register Busy

Bit 4 – PER PER Register Busy

Value	Description
0	Write synchronization for PER register is complete.
1	Write synchronization for PER register is ongoing.

Bit 3 – COUNT COUNT Register Busy

Value	Description
0	Read/write synchronization for COUNT register is complete.
1	Read/write synchronization for COUNT register is ongoing.

Bit 2 – **FREQCORR** FREQCORR Register Busy

Value	Description
0	Write synchronization for FREQCORR register is complete.
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – **ENABLE** Enable Bit Busy

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – **SWRST** Software Reset Bit Busy

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

25.8.8 Counter Value

Name: COUNT
Offset: 0x18
Reset: 0x0000
Property: RW

Table 25-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Counter Value

These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).

25.8.9 Counter Period

Name: PER
Offset: 0x1C
Reset: 0x0000
Property: RW

Table 25-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – PER[15:0] Counter Period

These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).

25.8.10 Compare n Value

Name: COMP
Offset: $0x20 + n \cdot 0x02$ [$n=0..5$]
Reset: 0x0000
Property: RW

Table 25-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	COMP[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COMP[15:0] Compare Value

The 16-bit value of COMP_n is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMP_n) is set on the next counter cycle.

25.8.11 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x00000000
Property: R

Table 25-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – COUNT[15:0] Count Timestamp Value

The 16-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs.

25.9 Register Summary - MODE2

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0	
0x00	CTRLA	15:8	CLOCKSYNC	GPTRST	BKTRST		PRESCALER[3:0]				
		7:0	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST	
0x02	CTRLB	15:8	SEPTO		ACTF[2:0]			DEBF[2:0]			
		7:0	DMAEN	RTCOUT	DEBASYNC	DEBMAJ	GP6EN	GP4EN	GP2EN	GP0EN	
0x04	EVCTRL	31:24								PERDEO	
		23:16								TAMPEVEI	
		15:8	OVFEO	TAMPEREO	ALARMEO[3:0]						
		7:0	PEREO[7:0]								
0x08	INTENCLR	15:8	OVF	TAMPER			ALARM[3:0]				
		7:0	PER[7:0]								
0x0A	INTENSET	15:8	OVF	TAMPER			ALARM[3:0]				
		7:0	PER[7:0]								
0x0C	INTFLAG	15:8	OVF	TAMPER			ALARM[3:0]				
		7:0	PER[7:0]								
0x0E ... 0x0F	Reserved										
0x10	SYNDBUSY	31:24									
		23:16	GP[7:0]								
		15:8	CLOCKSYNC	MASK[3:0]						ALARM[3]	
		7:0	ALARM[2:0]				CLOCK	FREQCORR	ENABLE	SWRST	
0x14 ... 0x17	Reserved										
0x18	CLOCK	31:24	YEAR[5:0]						MONTH[3:2]		
		23:16	MONTH[1:0]		DAY[4:0]				HOUR[4]		
		15:8	HOUR[3:0]				MINUTE[5:2]				
		7:0	MINUTE[1:0]		SECOND[5:0]						
0x1C ... 0x63	Reserved										
0x64	TIMESTAMP	31:24	YEAR[5:0]						MONTH[3:2]		
		23:16	MONTH[1:0]		DAY[4:0]				HOUR[4]		
		15:8	HOUR[3:0]				MINUTE[5:2]				
		7:0	MINUTE[1:0]		SECOND[5:0]						

25.9.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x0000
Property: RW

Table 25-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	CLOCKSYNC	GPTRST	BKTRST			PRESCALER[3:0]		
Access	RW	RW	RW		RW	RW	RW	RW
Reset	0	0	0		0	0	0	0

Bit	7	6	5	4	3	2	1	0
	MATCHCLR	CLKREP			MODE[1:0]		ENABLE	SWRST
Access	RW	RW			RW	RW	RW	W
Reset	0	0			0	0	0	0

Bit 15 – CLOCKSYNC Clock Read Synchronization Enable

The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

Value	Description
0	CLOCK read synchronization is disabled
1	CLOCK read synchronization is enabled

Bit 14 – GPTRST GP Registers Reset On Tamper Enable

Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is no synchronized.

Bit 13 – BKTRST BKUP Registers Reset On Tamper Enable

Bits 11:8 – PRESCALER[3:0] Prescaler

These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

Value	Name	Description
0x0	OFF	CLK_RTC_CNT = GCLK_RTC/1
0x1	DIV1	CLK_RTC_CNT = GCLK_RTC/1
0x2	DIV2	CLK_RTC_CNT = GCLK_RTC/2
0x3	DIV4	CLK_RTC_CNT = GCLK_RTC/4
0x4	DIV8	CLK_RTC_CNT = GCLK_RTC/8
0x5	DIV16	CLK_RTC_CNT = GCLK_RTC/16
0x6	DIV32	CLK_RTC_CNT = GCLK_RTC/32
0x7	DIV64	CLK_RTC_CNT = GCLK_RTC/64
0x8	DIV128	CLK_RTC_CNT = GCLK_RTC/128

Value	Name	Description
0x9	DIV256	CLK_RTC_CNT = GCLK_RTC/256
0xA	DIV512	CLK_RTC_CNT = GCLK_RTC/512
0xB	DIV1024	CLK_RTC_CNT = GCLK_RTC/1024

Bit 7 – MATCHCLR Clear on Match

This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	The counter is not cleared on a Compare/Alarm 0 match
1	The counter is cleared on a Compare/Alarm 0 match

Bit 6 – CLKREP Clock Representation

This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

Value	Description
0	24 Hour
1	12 Hour (AM/PM)

Bits 3:2 – MODE[1:0] Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

Value	Name	Description
0	COUNT32	Mode 0: 32-bit Counter
1	COUNT16	Mode 1: 16-bit Counter
2	CLOCK	Mode 2: Clock/Calendar

Bit 1 – ENABLE Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled
1	The peripheral is enabled

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.
This bit is not enable-protected.

Value	Description
0	There is not reset operation ongoing
1	The reset operation is ongoing

25.9.2 Control B

Name: CTRLB
Offset: 0x02
Reset: 0x0000
Property: RW

Table 25-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	SEPTO	ACTF[2:0]				DEBF[2:0]		
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bit	7	6	5	4	3	2	1	0
	DMAEN	RTCOUT	DEBASync	DEBMAJ	GP6EN	GP4EN	GP2EN	GP0EN
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – SEPTO Separate Tamper Outputs

Bits 14:12 – ACTF[2:0] Active Layer Frequency

Value	Name	Description
0x0	DIV2	CLK_RTC_OUT = CLK_RTC/2
0x1	DIV4	CLK_RTC_OUT = CLK_RTC/4
0x2	DIV8	CLK_RTC_OUT = CLK_RTC/8
0x3	DIV16	CLK_RTC_OUT = CLK_RTC/16
0x4	DIV32	CLK_RTC_OUT = CLK_RTC/32
0x5	DIV64	CLK_RTC_OUT = CLK_RTC/64
0x6	DIV128	CLK_RTC_OUT = CLK_RTC/128
0x7	DIV256	CLK_RTC_OUT = CLK_RTC/256

Bits 10:8 – DEBF[2:0] Debounce Frequency

Value	Name	Description
0x0	DIV2	CLK_RTC_DEB = CLK_RTC/2
0x1	DIV4	CLK_RTC_DEB = CLK_RTC/4
0x2	DIV8	CLK_RTC_DEB = CLK_RTC/8
0x3	DIV16	CLK_RTC_DEB = CLK_RTC/16
0x4	DIV32	CLK_RTC_DEB = CLK_RTC/32
0x5	DIV64	CLK_RTC_DEB = CLK_RTC/64
0x6	DIV128	CLK_RTC_DEB = CLK_RTC/128
0x7	DIV256	CLK_RTC_DEB = CLK_RTC/256

Bit 7 – DMAEN DMA Enable

Bit 6 – RTCOUT RTC Output Enable

Bit 5 – DEBASync Debouncer Asynchronous Enable

Bit 4 – DEBMAJ Debouncer Majority Enable

Bit 3 – GP6EN General Purpose 6 Enable

Bit 2 – GP4EN General Purpose 4 Enable

Bit 1 – GP2EN General Purpose 2 Enable

Bit 0 – GP0EN General Purpose 0 Enable

25.9.3 Event Control

Name: EVCTRL
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 25-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								PERDEO
Access								RW
Reset								0
Bit	23	22	21	20	19	18	17	16
								TAMPEVEI
Access								RW
Reset								0
Bit	15	14	13	12	11	10	9	8
	OVFEO	TAMPEREO				ALARMEO[3:0]		
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PEREO[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 24 – PERDEO Periodic Interval Daily Event Output Enable

Value	Description
0	Periodic Daily event is disabled and will not be generated.
1	Periodic Daily event is enabled and will be generated. The event occurs at the last second of each day depending on the CTRLA.CLKREP bit: - If CLKREP = 0, the event will occur at 23:59:59 - If CLKREP = 1, the event will occur at 11:59:59, PM = 1

Bit 16 – TAMPEVEI Tamper Event Input Enable

Value	Description
0	Tamper event input is disabled, and incoming events will be ignored.
1	Tamper event input is enabled, and all incoming events will capture the CLOCK value.

Bit 15 – OVFEO Overflow Event Output Enable

Value	Description
0	Overflow event is disabled and will not be generated.
1	Overflow event is enabled and will be generated for every overflow.

Bit 14 – TAMPEREO Tamper Event Output Enable

Value	Description
0	Tamper event output is disabled, and will not be generated
1	Tamper event output is enabled, and will be generated for every tamper input.

Bits 11:8 – ALARMEO[3:0] Alarm x Event Output Enable

Bits 7:0 – PEREO[7:0] Periodic Interval x Event Output Enable

25.9.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x08
Reset: 0x0000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Table 25-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			ALARM[3:0]			
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Enable

Bits 11:8 – ALARM[3:0] Alarm x Interrupt Enable

Bits 7:0 – PER[7:0] Periodic Interval x Interrupt Enable

25.9.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x0A
Reset: 0x0000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Table 25-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER				ALARM[3:0]		
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

Bit 14 – TAMPER Tamper Enable

Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

Value	Description
0	The Tamper interrupt is disabled.
1	The Tamper interrupt is enabled.

Bits 11:8 – ALARM[3:0] Alarm x Interrupt Enable

Bits 7:0 – PER[7:0] Periodic Interval x Enable

25.9.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x0C
Reset: 0x0000
Property: RW

Table 25-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	OVF	TAMPER			ALARM[3:0]			
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – OVF Overflow

This flag is cleared by writing a '1' to the flag.

This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the Overflow interrupt flag.

Bit 14 – TAMPER Tamper

This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

Bits 11:8 – ALARM[3:0] Alarm x

Bits 7:0 – PER[7:0] Periodic Interval x

25.9.7 Synchronization Busy Status

Name: SYNCBUSY
Offset: 0x10
Reset: 0x00000000
Property: R

Table 25-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	GP[7:0]							
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	CLOCKSYNC	MASK[3:0]						ALARM[3]
Reset	R	R	R	R	R			R
Reset	0	0	0	0	0			0
Bit	7	6	5	4	3	2	1	0
Access	ALARM[2:0]				CLOCK	FREQCORR	ENABLE	SWRST
Reset	R	R	R		R	R	R	R
Reset	0	0	0		0	0	0	0

Bits 23:16 – GP[7:0] General Purpose x Register Busy

Bit 15 – CLOCKSYNC Clock Synchronization Enable Bit Busy

Value	Description
0	Write synchronization for CTRLA.CLOCKSYNC bit is complete.
1	Write synchronization for CTRLA.CLOCKSYNC bit is ongoing.

Bits 14:11 – MASK[3:0] MASK x Register Busy

Bits 8:5 – ALARM[3:0] ALARM x Register Busy

Bit 3 – CLOCK CLOCK Register Busy

Value	Description
0	Read/write synchronization for CLOCK register is complete.
1	Read/write synchronization for CLOCK register is ongoing.

Bit 2 – FREQCORR FREQCORR Register Busy

Value	Description
0	Write synchronization for FREQCORR register is complete.

Value	Description
1	Write synchronization for FREQCORR register is ongoing.

Bit 1 – ENABLE Enable Bit Busy

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Bit Busy

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

25.9.8 Clock Value

Name: CLOCK
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 25-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]					MONTH[3:2]		
Access							RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]				HOUR[4]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year

The year offset with respect to the reference year (defined in software).
The year is considered a leap year if YEAR[1:0] is zero.

Bits 25:22 – MONTH[3:0] Month

1 – January
2 – February
...
12 – December

Bits 21:17 – DAY[4:0] Day

Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

Bits 16:12 – HOUR[4:0] Hour

When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

Bits 11:6 – MINUTE[5:0] Minute

0 – 59

Bits 5:0 – SECOND[5:0] Second
0 – 59

25.9.9 Timestamp

Name: TIMESTAMP
Offset: 0x64
Reset: 0x00000000
Property: R

Table 25-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]					MONTH[3:2]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]				HOUR[4]	
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year Timestamp Value

The year value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 25:22 – MONTH[3:0] Month Timestamp Value

The month value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 21:17 – DAY[4:0] Day Timestamp Value

The day value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 16:12 – HOUR[4:0] Hour Timestamp Value

The hour value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 11:6 – MINUTE[5:0] Minute Timestamp Value

The minute value is captured by the TIMESTAMP when a tamper condition occurs.

Bits 5:0 – SECOND[5:0] Second Timestamp Value

The second value is captured by the TIMESTAMP when a tamper condition occurs.

25.10 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x0D	Reserved									
0x0E	DBGCTRL	7:0								DBGRUN
0x0F ... 0x13	Reserved									
0x14	FREQCORR	7:0	SIGN	VALUE[6:0]						
0x15 ... 0x1F	Reserved									
0x20	ALARM - MODE2_ALARM Mode	31:24	YEAR[5:0]						MONTH[3:2]	
		23:16	MONTH[1:0]		DAY[4:0]				HOUR[4]	
		15:8	HOUR[3:0]				MINUTE[5:2]			
		7:0	MINUTE[1:0]		SECOND[5:0]					
0x24	MASK - MODE2_ALARM Mode	7:0						SEL[2:0]		
0x25 ... 0x3F	Reserved									
0x40	GP	31:24	GP[31:24]							
		23:16	GP[23:16]							
		15:8	GP[15:8]							
		7:0	GP[7:0]							
0x44 ... 0x5F	Reserved									
0x60	TAMPCTRL	31:24	DEBNC[7:0]							
		23:16	TAMLVL[7:0]							
		15:8	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
		7:0	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
0x64 ... 0x67	Reserved									
0x68	TAMPID	31:24	TAMPEVT							
		23:16								
		15:8								
		7:0	TAMPID[7:0]							
0x6C	TAMPCTRLB	31:24								
		23:16								
		15:8								
		7:0	ALSI[7:0]							
0x70 ... 0x7F	Reserved									
0x80	BKUP0	31:24	BKUP[31:24]							
		23:16	BKUP[23:16]							
		15:8	BKUP[15:8]							
		7:0	BKUP[7:0]							
...										
0xFC	BKUP31	31:24	BKUP[31:24]							
		23:16	BKUP[23:16]							
		15:8	BKUP[15:8]							
		7:0	BKUP[7:0]							

25.10.1 Debug Control

Name: DBGCTRL
Offset: 0x0E
Reset: 0x00
Property: RW

Table 25-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								RW
Reset								0

Bit 0 – DBGRUN Run During Debug

25.10.2 Frequency Correction

Name: FREQCORR
Offset: 0x14
Reset: 0x00
Property: RW

Table 25-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	SIGN	VALUE[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 7 – SIGN Correction Sign

Bits 6:0 – VALUE[6:0] Correction Value

25.10.3 Alarm n Value

Name: ALARM - MODE2_ALARM Mode
Offset: 0x20
Reset: 0x00000000
Property: RW

Table 25-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	YEAR[5:0]					MONTH[3:2]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MONTH[1:0]		DAY[4:0]				HOUR[4]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	HOUR[3:0]				MINUTE[5:2]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MINUTE[1:0]		SECOND[5:0]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:26 – YEAR[5:0] Year

Bits 25:22 – MONTH[3:0] Month

Bits 21:17 – DAY[4:0] Day

Bits 16:12 – HOUR[4:0] Hour

Value	Name	Description
0x00	AM	Morning hour
0x10	PM	Afternoon hour

Bits 11:6 – MINUTE[5:0] Minute

Bits 5:0 – SECOND[5:0] Second

25.10.4 Alarm n Mask

Name: MASK - MODE2_ALARM Mode
Offset: 0x24
Reset: 0x00
Property: RW

Table 25-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						SEL[2:0]		
Access								
Reset						0	0	0

Bits 2:0 – SEL[2:0] Alarm Mask Selection

Value	Name	Description
0x0	OFF	Alarm Disabled
0x1	SS	Match seconds only
0x2	MMSS	Match seconds and minutes only
0x3	HHMMSS	Match seconds, minutes, and hours only
0x4	DDHHMMSS	Match seconds, minutes, hours, and days only
0x5	MMDDHHMMSS	Match seconds, minutes, hours, days, and months only
0x6	YYMMDDHHMMSS	Match seconds, minutes, hours, days, months, and years

25.10.5 General Purpose

Name: GP
Offset: 0x40
Reset: 0x00000000
Property: RW

Table 25-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	GP[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	GP[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	GP[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	GP[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – GP[31:0] General Purpose

25.10.6 Tamper Control

Name: TAMPCTRL
Offset: 0x60
Reset: 0x00000000
Property: RW

Table 25-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DEBNC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TAMLVL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	IN7ACT[1:0]		IN6ACT[1:0]		IN5ACT[1:0]		IN4ACT[1:0]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	IN3ACT[1:0]		IN2ACT[1:0]		IN1ACT[1:0]		IN0ACT[1:0]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DEBNC[7:0] Debouncer Enable x

Bits 23:16 – TAMLVL[7:0] Tamper Level Select x

Bits 15:14 – IN7ACT[1:0] Tamper Input 7 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN7 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 13:12 – IN6ACT[1:0] Tamper Input 6 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN6 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 11:10 – IN5ACT[1:0] Tamper Input 5 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN5 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 9:8 – IN4ACT[1:0] Tamper Input 4 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN4 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 7:6 – IN3ACT[1:0] Tamper Input 3 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN3 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 5:4 – IN2ACT[1:0] Tamper Input 2 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN2 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 3:2 – IN1ACT[1:0] Tamper Input 1 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN1 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

Bits 1:0 – IN0ACT[1:0] Tamper Input 0 Action

Value	Name	Description
0x0	OFF	Off (Disabled)
0x1	WAKE	Wake and set Tamper flag
0x2	CAPTURE	Capture timestamp and set Tamper flag
0x3	ACTL	Compare IN0 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag

25.10.7 Tamper ID

Name: TAMPID
Offset: 0x68
Reset: 0x00000000
Property: RW

Table 25-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TAMPEVT							
Access	RW							
Reset	0							
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	TAMPID[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – TAMPEVT Tamper Event Detected

Bits 7:0 – TAMPID[7:0] Tamper Input x Detected

25.10.8 Tamper Control B

Name: TAMPCTRLB
Offset: 0x6C
Reset: 0x00000000
Property: RW

Table 25-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	ALSI[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ALSI[7:0] Active Layer Select Internal x

25.10.9 Backup

Name: BKUP
Offset: 0x80 + n*0x04 [n=0..31]
Reset: 0x00000000
Property: RW

Table 25-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BKUP[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BKUP[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BKUP[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BKUP[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BKUP[31:0] Backup

26. Direct Memory Access Controller (DMAC)

26.1 Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the [Block Diagram](#). The arbiter will grant one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has the following four bus interfaces:

- Data transfer bus – Used for performing the actual DMA transfer.
- AHB/APB Bridge bus – Used when writing and reading the I/O registers of the DMAC.
- Descriptor fetch bus – Used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- Write-back bus – Used to write the transfer descriptor back to SRAM.

All buses are AHB host interfaces except the AHB/APB Bridge bus, which is an APB client interface.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

26.2 Features

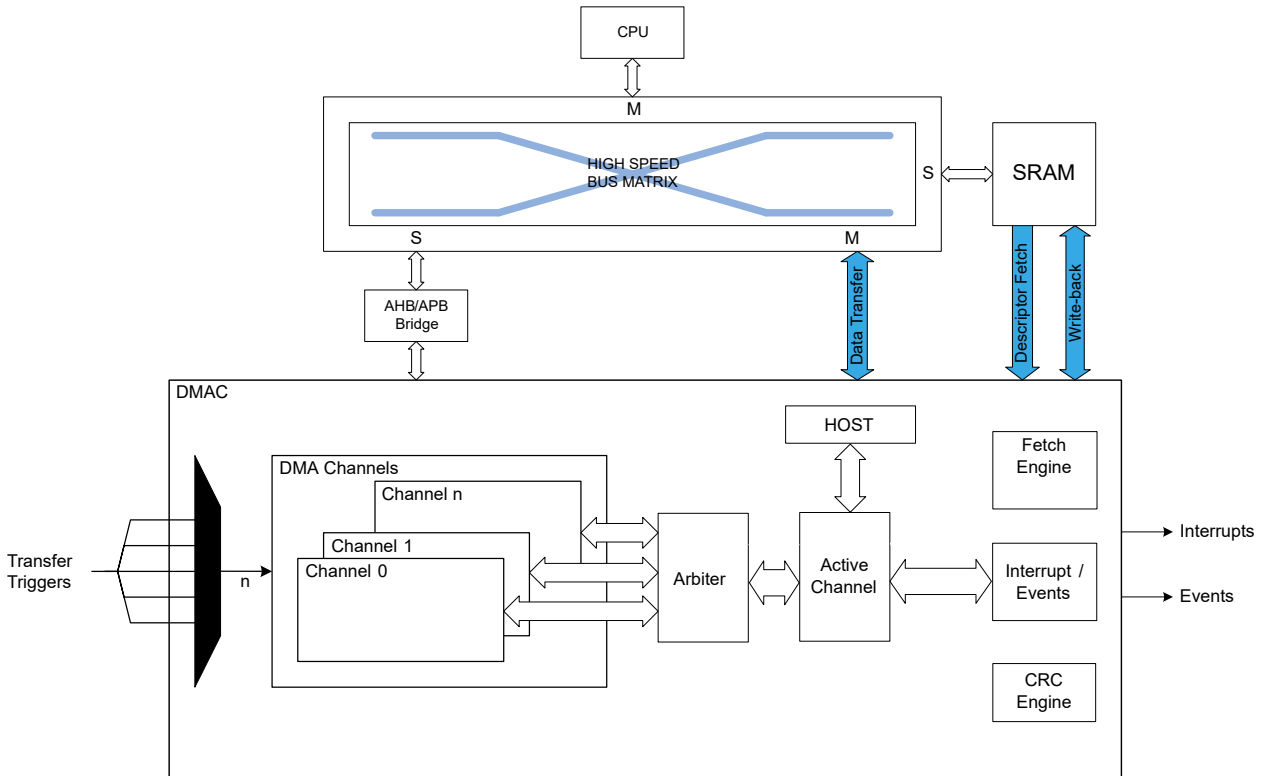
The following are key features of the DMAC module:

- Data transfer from:
 - Peripheral-to-peripheral
 - Peripheral-to-memory
 - Memory-to-peripheral
 - Memory-to-memory
- Transfer trigger sources
 - Software
 - Events from Event System
 - Dedicated requests from peripherals
- SRAM based transfer descriptors
 - Single transfer using one descriptor
 - Multi-buffer or circular buffer modes by linking multiple descriptors
- 8 channels

- Enable 8 independent transfers
 - Automatic descriptor fetch for each channel
 - Suspend/resume operation support for each channel
- Flexible arbitration scheme
 - 4 configurable priority levels for each channel
 - Fixed or round-robin priority scheme within each priority level
- From 1 KB to 256 KB data transfer in a single block transfer
- Multiple addressing modes
 - Static
 - Configurable increment scheme
- Optional interrupt generation
 - On block transfer complete
 - On error detection
 - On channel suspend
- 4 event inputs
 - One event input for each of the 4 least significant DMA channels
 - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
 - Can be selected to suspend or resume channel operation
- 4 event outputs
 - One output event for each of the 4 least significant DMA channels
 - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
 - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable
 - CRC-16 (CRC-CCITT)
 - CRC-32 (IEEE[®] 802.3)

26.3 Block Diagram

Figure 26-1. DMAC Block Diagram



26.4 Functional Description

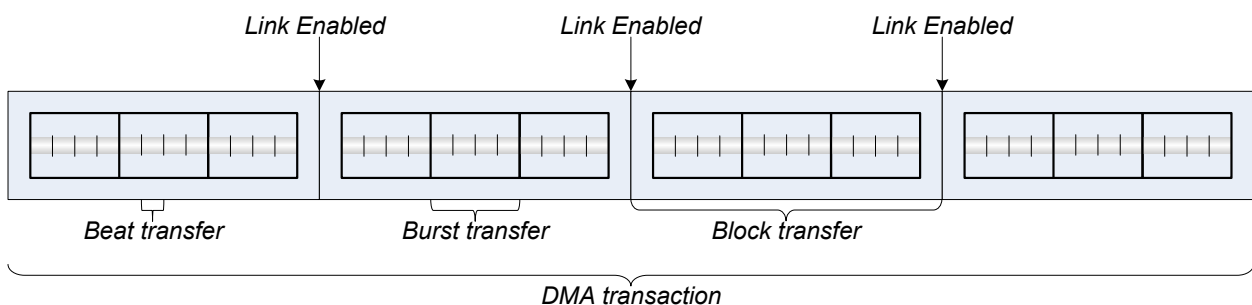
26.4.1 Principle of Operation

The DMAC consists of a DMA module and a CRC module.

26.4.1.1 DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure shows the relationship between the different transfer sizes:

Figure 26-2. DMA Transfer Sizes



- Beat transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE).
- Block transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1k to 64k beats. A block transfer can be interrupted.

- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in SRAM. For additional information on the transfer descriptor, refer to the [Transfer Descriptors](#).

The figure above shows several block transfers linked together, which are called linked descriptors. For additional information about linked descriptors, refer to the [Linked Descriptors](#).

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent of the configuration, the DMA channel will either be suspended or disabled.

26.4.1.2 CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to [CRC Operation](#) for details.

26.4.2 Basic Operation

26.4.2.1 Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE = 0):

- The Descriptor Base Memory Address register (BASEADDR)
- The Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE = 0 and CTRL.CRCENABLE = 0):

- The Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE = 0):

- The Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- The Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE = 0):

- The CRC Control register (CRCCTRL)
- The CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the 'Enable-Protected' property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address register (BASEADDR).
- The SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address register (WRBADDR).
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLEN x = 1)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
 - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register.
 - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT).
 - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC).
- Transfer Descriptor
 - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE).
 - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID).
 - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register.
 - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register.
 - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register.

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must be selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC).
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY).
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control register (CRCCTRL.CRCBEATSIZE).

26.4.2.2 Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the reset to take effect.

26.4.2.3 Transfer Descriptors

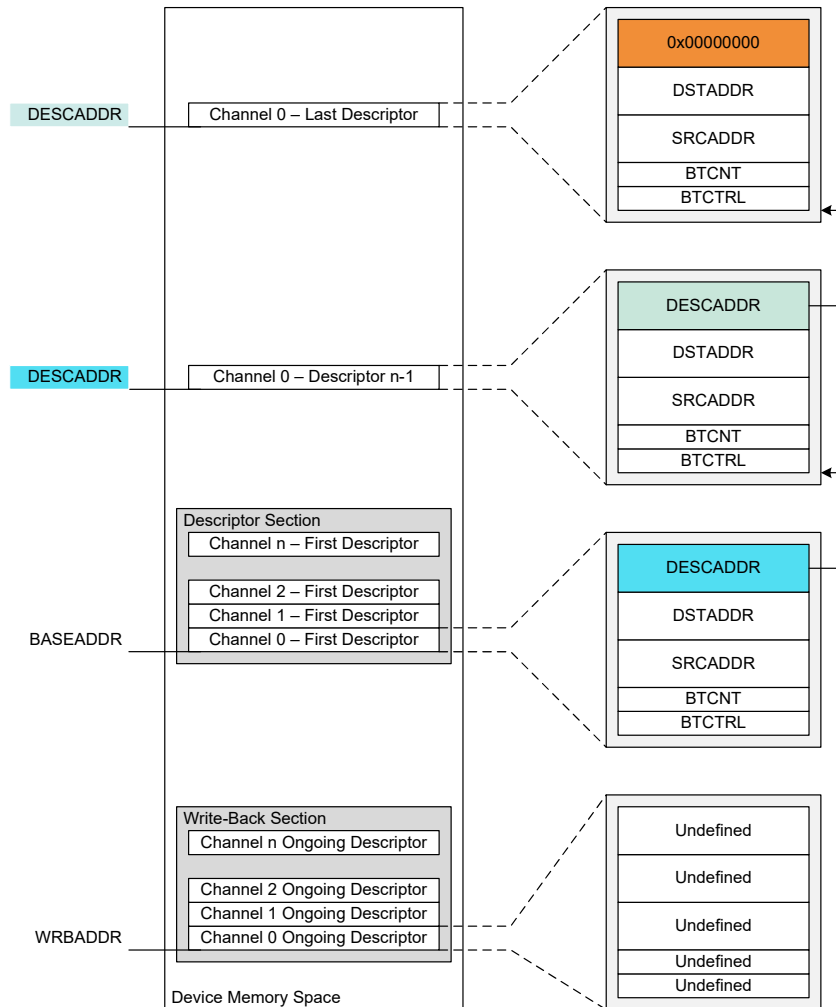
Together with the channel configurations the transfer descriptors decides how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one), and receives a transfer trigger, its first transfer descriptor has to be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel 0 (see figure below), all first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For additional information on linked descriptors, refer to the [Linked Descriptors](#).

The write-back memory section is the section where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel 0. All ongoing transfer descriptors will be stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel 0. For additional information on linked descriptors, refer to the [Linked Descriptors](#).

Figure 26-3. Memory Sections



The size of the descriptor and write-back memory sections is dependent on the number of the most significant enabled DMA channel m , as shown below:

$$Size = 128\text{bits} \cdot (m + 1)$$

For memory optimization, it is recommended to use the less significant DMA channels, if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share memory section ($BASEADDR = WRBADDR$). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less SRAM.

26.4.2.4 Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel x bit in the Pending Channels registers (PENDCH.PENDCH x) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The active channel is the DMA channel being granted access to perform its next transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit (PENDCH.PENDCH x) will be cleared. See also the following figure.

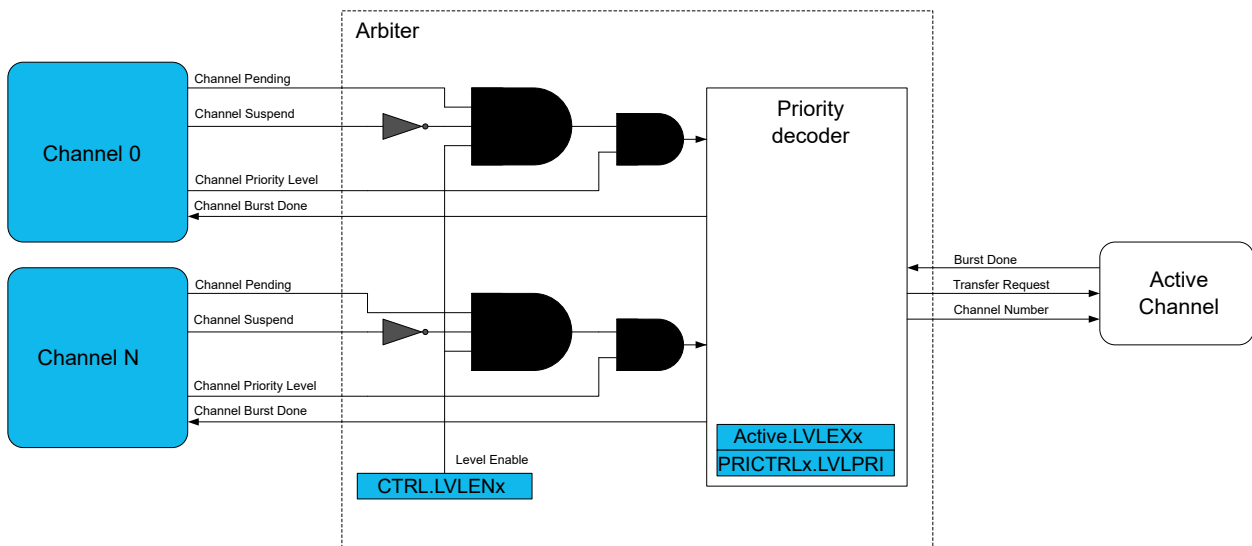
If the upcoming transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register (BUSYCH.BUSYCHx = 1) will be set, and it will remain '1' for the subsequent granted transfers.

When the channel has performed its granted transfers it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended, or disabled. This depends on the channel and block transfer configuration. If the DMA channel is fed into the queue of channels with pending transfers, the corresponding BUSYCH.BUSYCHx will remain '1'. If the DMA channel is set to wait for a new transfer trigger, suspended, or disabled, the corresponding BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE = 0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

Figure 26-4. Arbiter Overview



Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

Each DMA channel supports a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register (CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (CTRL.LVLENx = 1).

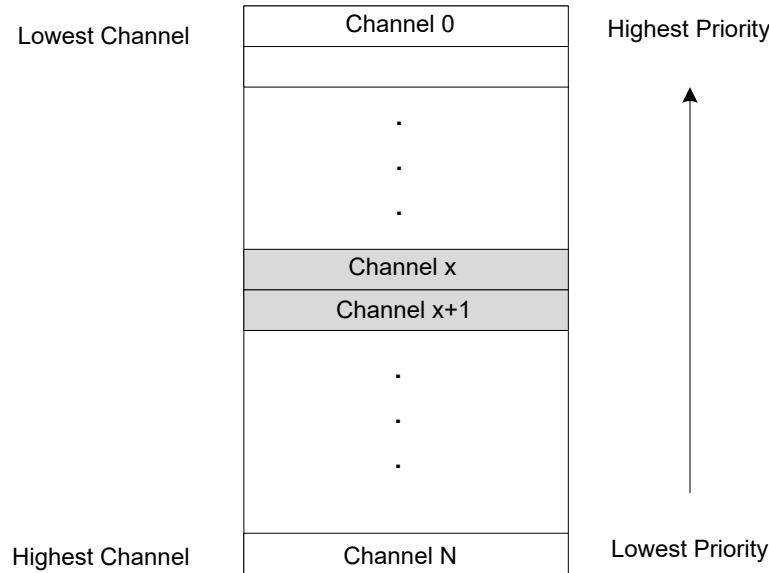
Within each priority level the DMAC's arbiter can be configured to prioritize statically or dynamically:

Static Arbitration within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control register (PRICTRL.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration there is a risk of

high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

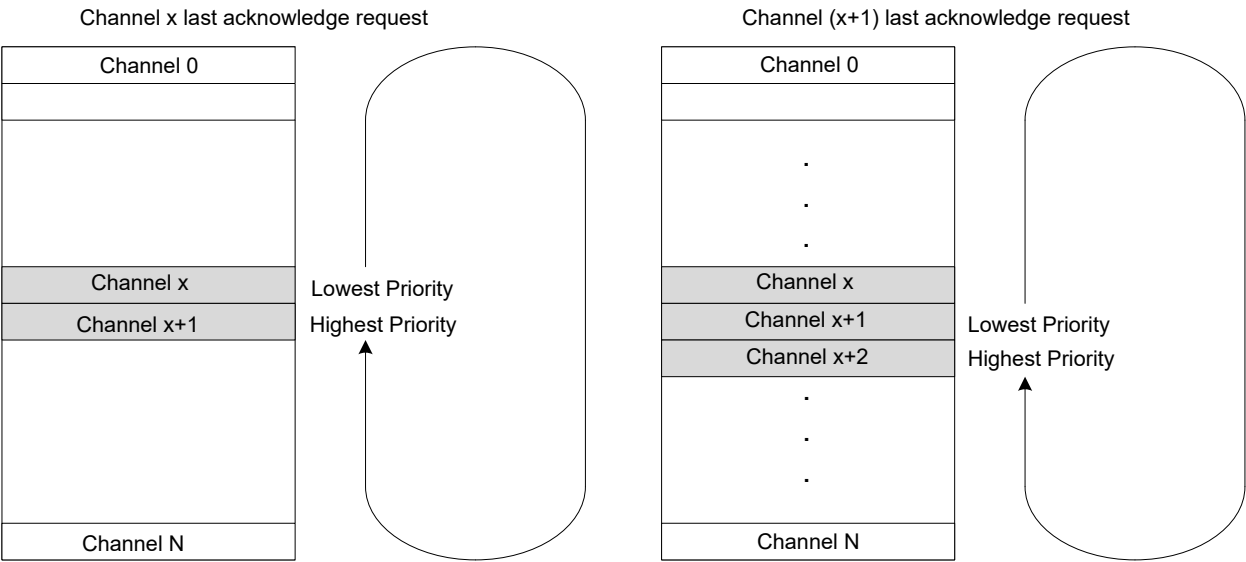
Figure 26-5. Static Priority Scheduling



Dynamic Arbitration within a priority level is selected by writing a '1' to PRICTRL.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in Dynamic (Round-Robin) Priority Scheduling. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control register (PRICTRL.LVLPRIx) for the corresponding priority level.

Figure 26-6. Dynamic (Round-Robin) Priority Scheduling



26.4.2.5 Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to DMA Block Diagram section) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on *Addressing*.

The arbitration procedure is performed after each transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID = 0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

26.4.2.6 Transfer Triggers and Actions

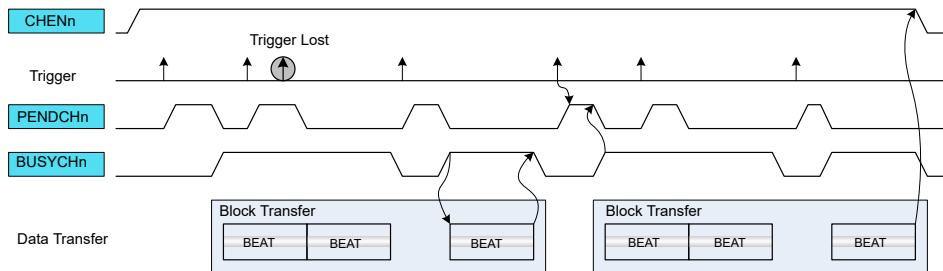
A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel Control B register (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. If the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a beat transfer (CHCTRLB.TRIGACT = 0x2) or transaction transfer (CHCTRLB.TRIGACT = 0x3) instead of a block transfer (CHCTRLB.TRIGACT = 0x0).

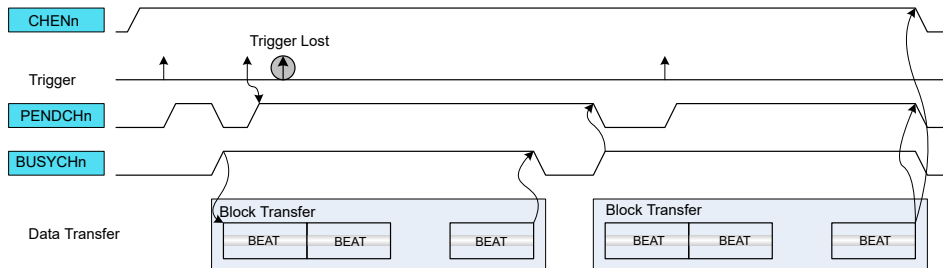
Figure 26-7 shows an example where triggers are used with two linked block descriptors.

Figure 26-7. Trigger Action and Transfers

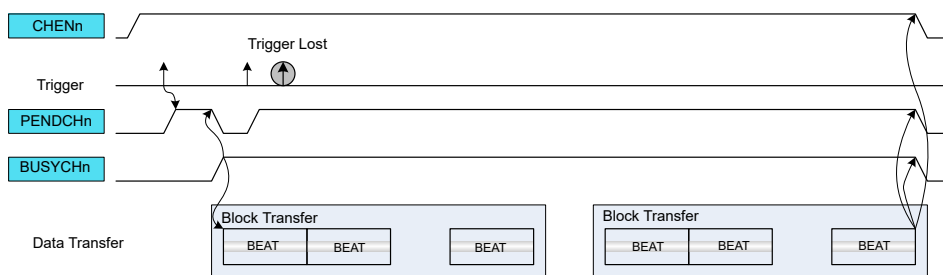
Beat Trigger Action



Block Trigger Action



Transaction Trigger Action



If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUS.PEND = 1), and the new transfer can start after the ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel Status register (CHSTATUS.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

26.4.2.7 Addressing

Each block transfer must have both a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (DSTADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC = 1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in

the Block Transfer Control register (BTCTRL.STEPSEL = 1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL = 0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC = 1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL = 1:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1) \cdot 2^{\text{STEPSIZE}}$$

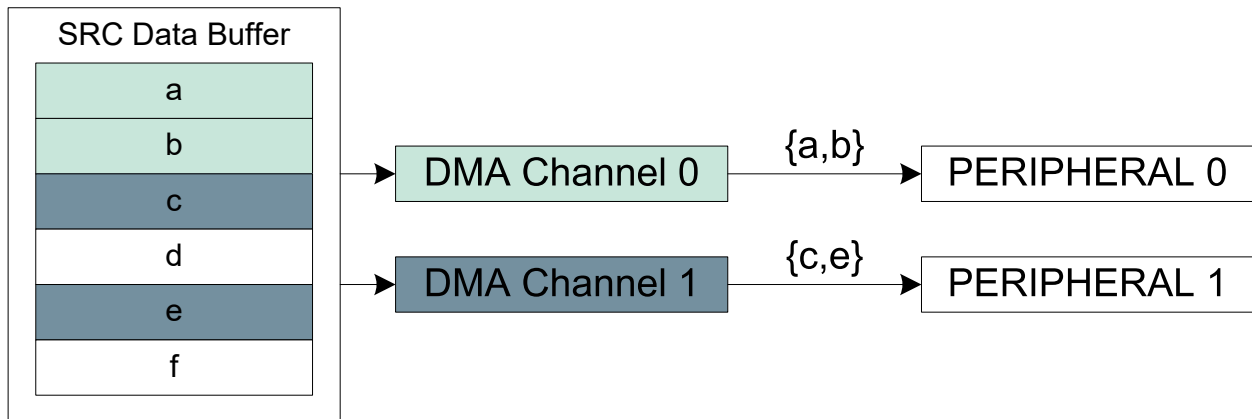
If BTCTRL.STEPSEL = 0:

$$\text{SRCADDR} = \text{SRCADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1)$$

- SRCADDR_{START} is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC = 1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC = 1, BTCTRL.STEPSEL = 1, and BTCTRL.STEPSIZE = 0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC = 0).

Figure 26-8. Source Address Increment



Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC = 1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL = 0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL = 1, the step size for the destination incrementation will be the size of one beat.

When the destination address incrementation is configured (BTCTRL.DSTINC = 1), DSTADDR must be set and calculated as follows:

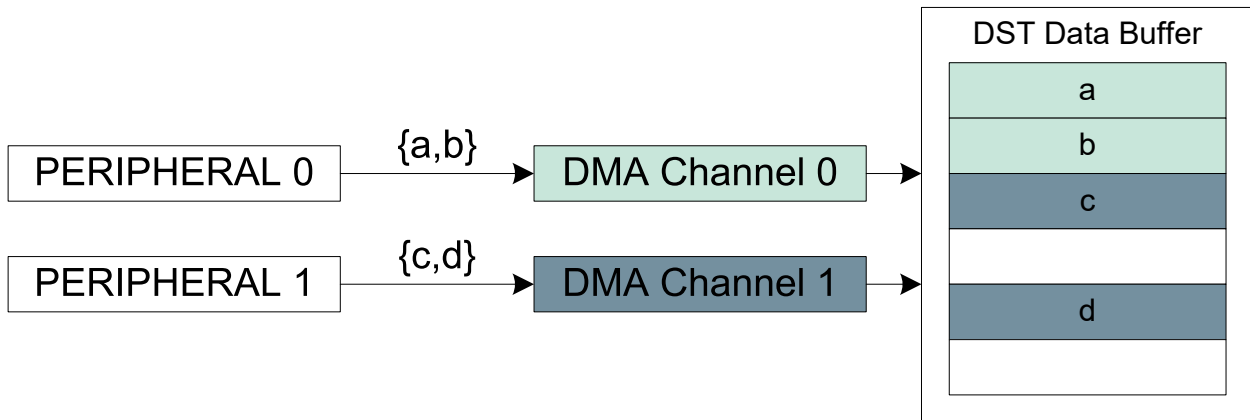
$\text{DSTADDR} = \text{DSTADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1) \cdot 2^{\text{STEPSIZE}}$	where BTCTRL.STEPSEL is zero
$\text{DSTADDR} = \text{DSTADDR}_{\text{START}} + \text{BTCNT} \cdot (\text{BEATSIZE} + 1)$	where BTCTRL.STEPSEL is one

- DSTADDR_{START} is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer

- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC = 1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC = 1, BTCTRL.STEPSEL = 0, and BTCTRL.STEPSIZE = 0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC = 0).

Figure 26-9. Destination Address Increment



26.4.2.8 Error Handling

If a bus error is received from an AHB client during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID = 0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR = 0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

26.4.3 Additional Features

26.4.3.1 Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consists of several block transfers it is done with the help of linked descriptors.

Figure 26-3 illustrates how linked descriptors work. When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor, which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor from the Next Descriptor Address (DESCADDR) register is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR = 0x00000000, the transaction is terminated. For additional information on how the next descriptor is fetched from SRAM, refer to the section *Data Transmission*.

26.4.3.1.1 Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in SRAM, with DESCADDR = 0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

26.4.3.1.2 Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in SRAM to configure a new descriptor.
4. Configure the new descriptor:
 - Set the next descriptor address in the Next Descriptor Address (DESCADDR) register
 - Set the destination address in the Block Transfer Destination Address (DSTADDR) register
 - Set the source address in the Block Transfer Source Address (SRCADDR) register
 - Configure the Block Transfer Control (BTCTRL) register, including
 - Optionally enable the Suspend block action
 - Set the descriptor VALID bit
5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read Next Descriptor Address (DESCADDR) register from the Write-Back memory.
 - If the DMA has not already fetched the descriptor which requires changes (i.e., DESCADDR is wrong):
 - Update the Next Descriptor Address (DESCADDR) register location of the descriptor from the List
 - Optionally clear the Suspend block action
 - Set the descriptor VALID bit to '1'
 - Optionally enable the Resume software command
 - If the DMA is executing the same descriptor as the one which requires changes:
 - Set the Channel Suspend software command and wait for the Suspend interrupt
 - Update the Next Descriptor Address (DESCADDR) register in the write-back memory
 - Clear the interrupt sources and set the Resume software command
 - Update the Next Descriptor Address (DESCADDR) register location of the descriptor from the List
 - Optionally clear the Suspend block action
 - Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

26.4.3.1.3 Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow these steps:
 - a. Set the descriptor A VALID bit to '0'.
 - b. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
 - c. Set the DESCADDR value of descriptor C to point to descriptor B.
 - d. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:
 - a. Apply the software suspend command to the channel.
 - b. Perform Step 2.1 through Step 2.4.

- c. Apply the software resume command to the channel.

26.4.3.2 Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP = 1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID = 0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register (CHASTATUS.FERR) will be set.

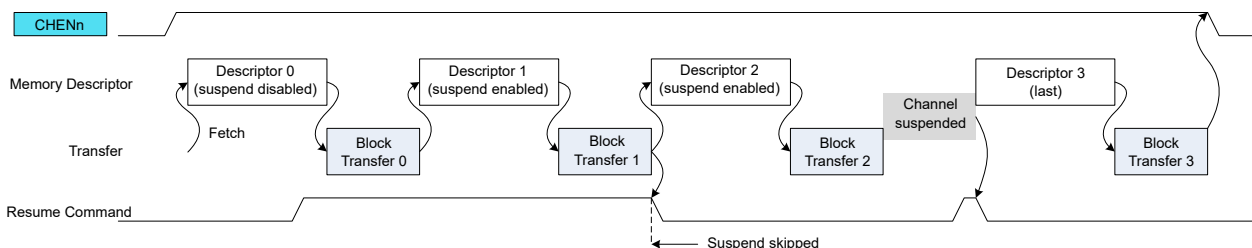
Note: Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

For additional information on transfer descriptors, refer to the [Transfer Descriptors](#).

26.4.3.3 Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.

Figure 26-10. Channel Suspend/Resume Operation



26.4.3.4 Event Input Actions

The event input actions are available only on the 4 least significant DMA channels. For details on channels with event input support, refer to the Chapter Event System (EVSYS).

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to the section Events.

Table 26-1. Event Input Action

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
None	NOACT	-
Normal Transfer	TRIG	DISABLE

Table 26-1. Event Input Action (continued)

Action	CHCTRLB.EVACT	CHCTRLB.TRGSRC
Conditional Transfer on Strobe	TRIG	any peripheral
Conditional Transfer	CTRIG	
Conditional Block Transfer	CBLOCK	
Channel Suspend	SUSPEND	
Channel Resume	RESUME	
Skip Next Block Suspend	SSKIP	

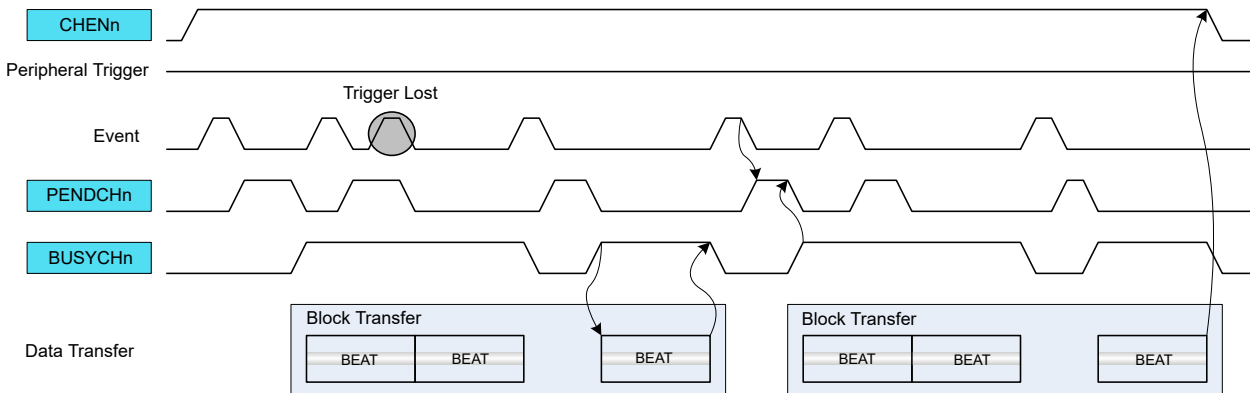
Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

Figure 26-11. Beat Event Trigger Action



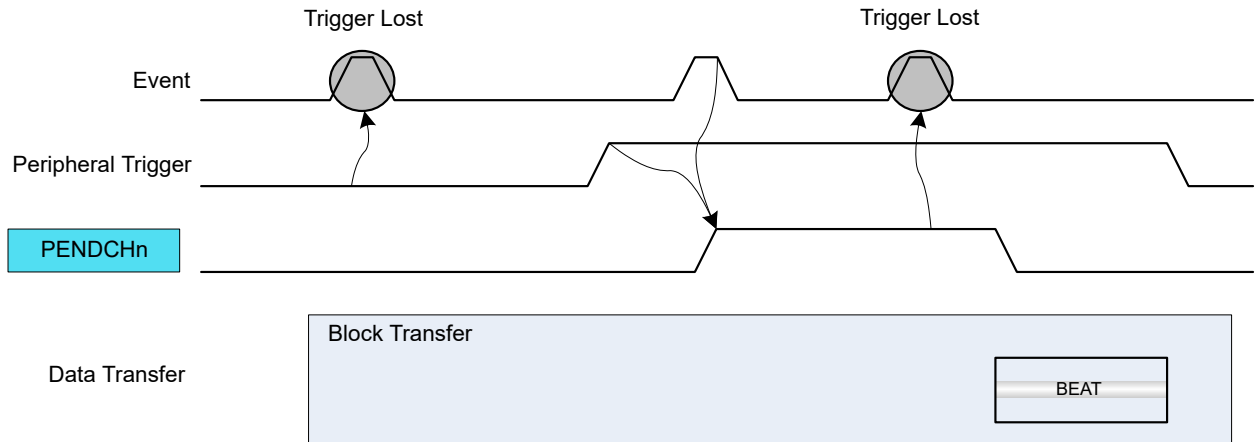
Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, e.g. for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (i.e. the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both Channel Status register (CHSTATUS.PEND) and Pending Channels register (PENDCH.PENDCHn) are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

Figure 26-12. Periodic Event with Beat Peripheral Triggers



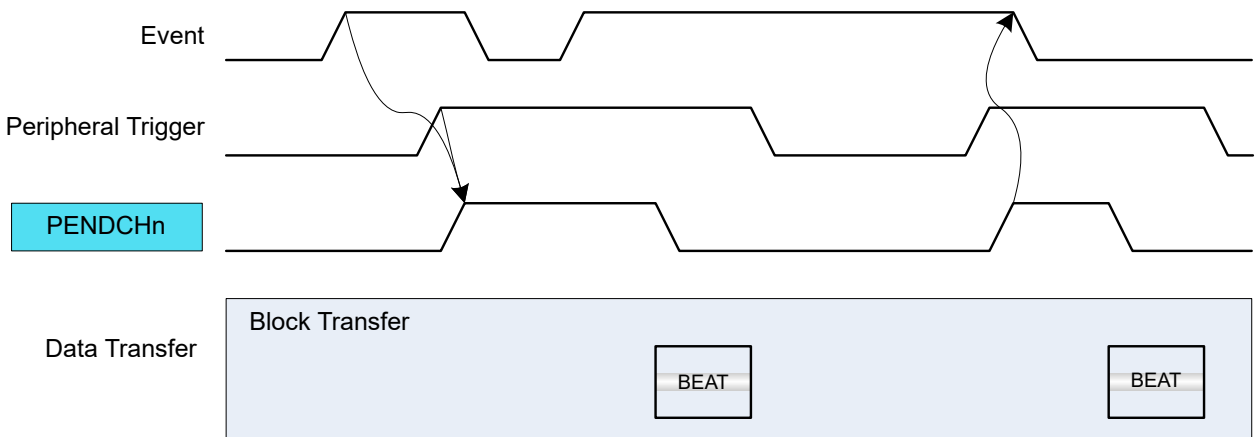
Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set in Channel Status register (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

Figure 26-13. Conditional Event with Beat Peripheral Triggers



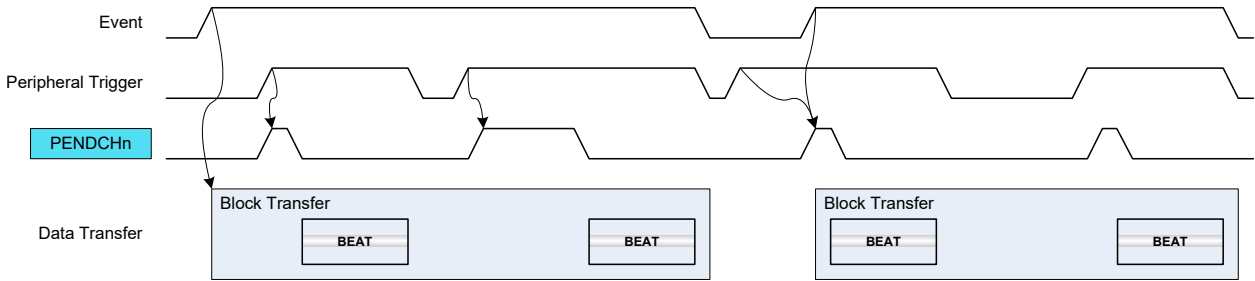
Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

Figure 26-14. Conditional Block Transfer with Beat Peripheral Triggers



Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For further details on Channel Suspend, refer to Channel Support.

Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. For further details refer to Channel Support.

Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

26.4.3.5 Event Output Selection

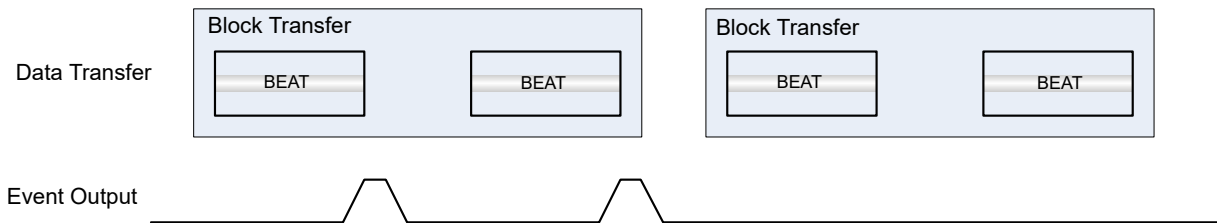
Event output selection is available only for the 4 least significant DMA channels. The pulse width of an event output from a channel is one AHB clock cycle.

The output of channel events is enabled by writing a '1' to the Channel Event Output Enable bit in the Control B register (CHCTRLB.EVOE). The event output cause is selected by writing to the Event Output Selection bits in the Block Transfer Control register (BTCTRL.EVOSEL). It is possible to generate events after each block transfer (BTCTRL.EVOSEL = 0x1) or beat transfer (BTCTRL.EVOSEL = 0x3). To enable an event being generated when a transaction is complete, the block event selection must be set in the last transfer descriptor only.

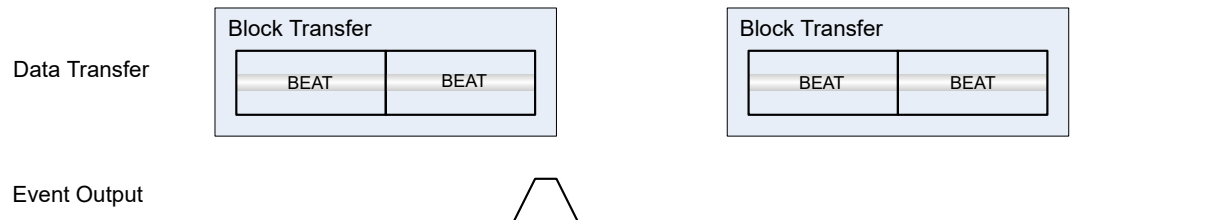
The figure [Figure 26-15](#) shows an example where the event output generation is enabled in the first block transfer, and disabled in the second block.

Figure 26-15. Event Output Generation

Beat Event Output



Block Event Output



26.4.3.6 Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE = 0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE = 0) when the entire DMAC module is disabled.

26.4.3.7 CRC Operation

A cyclic redundancy check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16-bit or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation. If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is $\leq n$ bits in length, and will detect the fraction $1-2^{-n}$ of all longer error bursts.

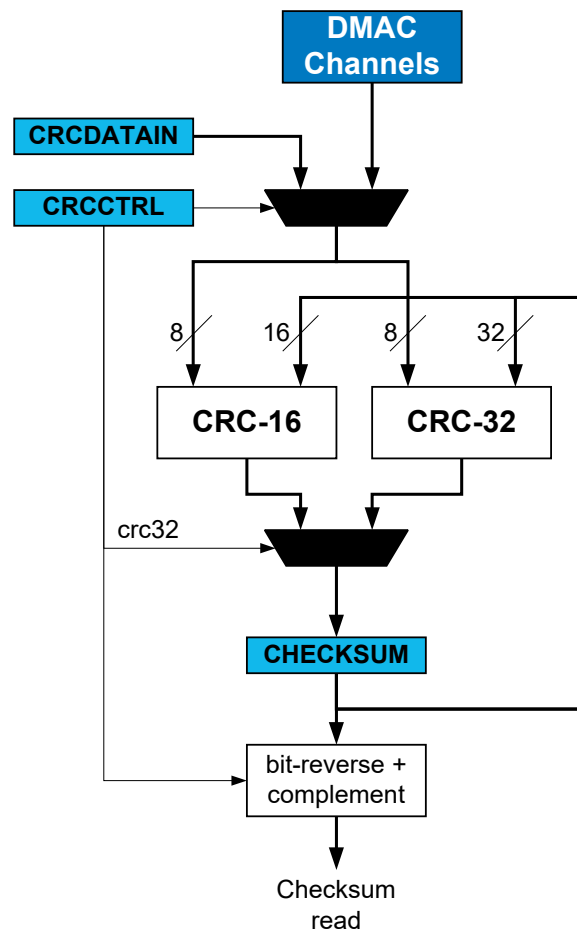
- CRC-16:

- Polynomial: $x^{16} + x^{12} + x^5 + 1$
- Hex value: 0x1021
- CRC-32:
 - Polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
 - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in Figure 26-16.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-bit, 16-bit, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRC Data Input (CRCDATAIN) register and the CRC engine will operate on the input data in a byte by byte manner.

Figure 26-16. CRC Generator Block Diagram



CRC on DMA data

CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will

continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the CRC Data Input (CRCDATAIN) register in the CRC engine.

CRC using the I/O interface Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected.

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRC Data Input (CRCDATAIN) register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRC Data Input (CRCDATAIN) register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

26.4.4 Interrupts

The DMAC channels have the following interrupt sources:

- **Transfer Complete (TCMPL):** Indicates that a block transfer is completed on the corresponding channel. Refer to *Data Transmission* for details.
- **Transfer Error (TERR):** Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to *Error Handling* for details.
- **Channel Suspend (SUSP):** Indicates that the corresponding channel has been suspended. Refer to *Channel Suspend* and *Data Transmission* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET = 1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR = 1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear interrupt flags. Interrupt requests for DMA channels 0 to 3 are independently connected to the Nested Vectored Interrupt Controller. Interrupt requests for DMA channels 4 to 7 are ORed together on system level to generate one combined interrupt request to the Nested Vectored Interrupt Controller.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags.



Important: INTPEND is dedicated to channels which are multiplexed on a single interrupt line (Channels 4 to 7).

26.4.5 Events

The DMAC can generate the following output events:

- Channel (CH0-3): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to the [Event Output Selection](#) for details.

Setting the Channel Control B Event Output Enable bit (CHCTRLB.EVOE = 1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHCTRLB.EVOE = 0 disables the corresponding output event.

The DMAC can take the following actions on an input event (CH0-3):

- Transfer and Periodic Transfer Trigger (TRIG): Normal transfer or periodic transfers on peripherals are enabled.
- Conditional Transfer Trigger (CTRIG): Conditional transfers on peripherals are enabled.
- Conditional Block Transfer Trigger (CBLOCK): Conditional block transfers on peripherals are enabled.
- Channel Suspend Operation (SUSPEND): Suspend a channel operation.
- Channel Resume Operation (RESUME): Resume a suspended channel operation.
- Skip Next Block Suspend Action (SSKIP): Skip the next block suspend transfer condition.
- Increase Priority (INCPRI): Increase channel priority.

Setting the Channel Control B Event Input Enable bit (CHCTRLB.EVIE = 1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For additional information on event input actions, refer to the [Event Input Actions](#).

Note: Event input and outputs are not available for every channel. Refer to the [Features](#) for additional information.

26.4.6 Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in Standby mode, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

For channels with CHCTRLA.RUNSTDBY = 0, it is up to the software to stop DMA transfers on these channels and wait for completion before going to Standby mode using the following steps:

1. Suspend the DMAC channels for which CHCTRLA.RUNSTDBY = 0.
2. Check the SYNCBUSY bits of registers accessed by the DMAC channels being suspended.
3. Go to sleep.
4. When the device wakes up, resume the suspended channels.

26.4.7 Debug Operation

When the CPU is halted in Debug mode, the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to the DBGCTRL for details.

26.5 Peripheral Dependencies

Table 26-2. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
DMAC	0x44802000	42 : TCMPL0/ SUSPO/TERR0 43 : TCMPL1/ SUSP1/TERR1 44 : TCMPL2/ SUSP2/TERR2 45 : TCMPL3/ SUSP3/TERR3 46 : TCMPL4/ SUSP4/TERR4/ TCMPL5/ SUSP5/TERR5/ TCMPL6/ SUSP6/TERR6/ TCMPL7/ SUSP7/TERR7	AHB: CLKMSK[0] MASK28 APB: CLKMSK[0] MASK29	---	21 STATUS0 PERID21		<u>Generators:</u> CH0: CHANNEL[37] CH1: CHANNEL[38] CH2: CHANNEL[39] CH3: CHANNEL[40] <u>Users:</u> CH0: USER[6],Resynch CH1: USER[7],Resynch CH2: USER[8],Resynch CH3: USER[9],Resynch	PD_CORE_SW

26.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0	
0x00	CTRL	15:8	LVLEN[7:0]								
		7:0						CRCENABLE	DMAENABLE	SWRST	
0x02	CRCCTRL	15:8			CRCSRC[5:0]						
		7:0					CRCPOLY[1:0]		CRCBEATSIZE[1:0]		
0x04	CRCDATAIN	31:24	CRCDATAIN[31:24]								
		23:16	CRCDATAIN[23:16]								
		15:8	CRCDATAIN[15:8]								
		7:0	CRCDATAIN[7:0]								
0x08	CRCCHKSUM	31:24	CRCCHKSUM[31:24]								
		23:16	CRCCHKSUM[23:16]								
		15:8	CRCCHKSUM[15:8]								
		7:0	CRCCHKSUM[7:0]								
0x0C	CRCSTATUS	7:0							CRCZERO	CRCBUSY	
0x0D	DBGCTRL	7:0								DBGRUN	
0x0E	QOSCTRL	7:0			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]		
0x0F	Reserved										
0x10	SWTRIGCTRL	31:24	SWTRIG[31:24]								
		23:16	SWTRIG[23:16]								
		15:8	SWTRIG[15:8]								
		7:0	SWTRIG[7:0]								
0x14	PRICTRL0	31:24	RRLVLEN3			LVLPRI3[4:0]					
		23:16	RRLVLEN2			LVLPRI2[4:0]					
		15:8	RRLVLEN1			LVLPRI1[4:0]					
		7:0	RRLVLEN0			LVLPRI0[4:0]					
0x18	PRICTRL1	31:24	RRLVLEN7			LVLPRI7[4:0]					
		23:16	RRLVLEN6			LVLPRI6[4:0]					
		15:8	RRLVLEN5			LVLPRI5[4:0]					
		7:0	RRLVLEN4			LVLPRI4[4:0]					
0x1C ... 0x1F	Reserved										
0x20	INTPEND	15:8	PEND	BUSY	FERR			SUSP	TCMPL	TERR	
		7:0				ID[4:0]					
0x22 ... 0x23	Reserved										
0x24	INTSTATUS	31:24	CHINT[31:24]								
		23:16	CHINT[23:16]								
		15:8	CHINT[15:8]								
		7:0	CHINT[7:0]								
0x28	BUSYCH	31:24	BUSYCH[31:24]								
		23:16	BUSYCH[23:16]								
		15:8	BUSYCH[15:8]								
		7:0	BUSYCH[7:0]								
0x2C	PENDCH	31:24	PENDCH[31:24]								
		23:16	PENDCH[23:16]								
		15:8	PENDCH[15:8]								
		7:0	PENDCH[7:0]								
0x30	ACTIVE	31:24	BTCNT[15:8]								
		23:16	BTCNT[7:0]								
		15:8	ABUSY			ID[4:0]					
		7:0	LVLEX[7:0]								
0x34	BASEADDR	31:24	BASEADDR[31:24]								
		23:16	BASEADDR[23:16]								
		15:8	BASEADDR[15:8]								
		7:0	BASEADDR[7:0]								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	WRBADDR	31:24	WRBADDR[31:24]							
		23:16	WRBADDR[23:16]							
		15:8	WRBADDR[15:8]							
		7:0	WRBADDR[7:0]							
0x3C ... 0x3E	Reserved									
0x3F	CHID	7:0	ID[4:0]							
0x40	CHCTRLA	7:0		RUNSTDBY					ENABLE	SWRST
0x41 ... 0x43	Reserved									
0x44	CHCTRLB	31:24	CMD[1:0]							
		23:16	TRIGACT[1:0]							
		15:8	TRIGSRC[7:0]							
		7:0	LVL[2:0]			EVOE	EVIE	EVACT[2:0]		
0x48 ... 0x4B	Reserved									
0x4C	CHINTENCLR	7:0						SUSP	TCMPL	TERR
0x4D	CHINTENSET	7:0						SUSP	TCMPL	TERR
0x4E	CHINTFLAG	7:0						SUSP	TCMPL	TERR
0x4F	CHSTATUS	7:0						FERR	BUSY	PEND

26.6.1 Control

Name: CTRL
Offset: 0x00
Reset: 0x0000
Property: RW

Bit	15	14	13	12	11	10	9	8
	LVLEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						CRCENABLE	DMAENABLE	SWRST
Access						RW	RW	RW
Reset						0	0	0

Bits 15:8 – LVLEN[7:0] Priority Level x Enable

Bit 2 – CRCENABLE CRC Enable

Bit 1 – DMAENABLE DMA Enable

Bit 0 – SWRST Software Reset

26.6.2 CRC Control

Name: CRCCTRL
Offset: 0x02
Reset: 0x0000
Property: RW

Bit	15	14	13	12	11	10	9	8
	CRCSRC[5:0]							
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
					CRCPOLY[1:0]		CRCBEATSIZE[1:0]	
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 13:8 – CRCSRC[5:0] CRC Input Source

These bits select the input source for generating the CRC, as shown in the table below. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY Status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

Value	Name	Description
	CHN	DMA channel n
0x00	NOACT	No action
0x01	IO	I/O interface

Bits 3:2 – CRCPOLY[1:0] CRC Polynomial Type

These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

Value	Name	Description
0x0	CRC16	CRC-16 (CRC-CCITT)
0x1	CRC32	CRC32 (IEEE 802.3)

Bits 1:0 – CRCBEATSIZE[1:0] CRC Beat Size

These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	WORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer

26.6.3 CRC Data Input

Name: CRCDATAIN
Offset: 0x04
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	CRCDATAIN[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCDATAIN[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCDATAIN[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCDATAIN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCDATAIN[31:0] CRC Data Input
These bits store the data for which the CRC checksum is computed. A new CRC Checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.

26.6.4 CRC Checksum

Name: CRCCHKSUM
Offset: 0x08
Reset: 0x00000000
Property: RW

The CRCCHKSUM represents the 16-bit or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

Bit	31	30	29	28	27	26	25	24
	CRCCHKSUM[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCCHKSUM[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCCHKSUM[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCCHKSUM[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCCHKSUM[31:0] CRC Checksum

These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.

26.6.5 CRC Status

Name: CRCSTATUS
Offset: 0x0C
Reset: 0x00
Property: RW

Bit	7	6	5	4	3	2	1	0
							CRCZERO	CRCBUSY
Access							R	RW
Reset							0	0

Bit 1 – CRCZERO CRC Zero

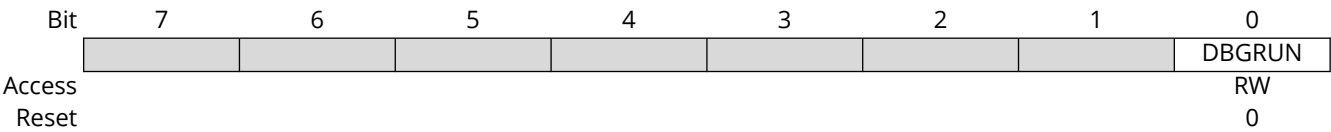
This bit is cleared when a new CRC source is selected.
This bit is set when the CRC generation is complete and the CRC Checksum is zero.
When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

Bit 0 – CRCBUSY CRC Module Busy

This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.
This bit is set when a source configuration is selected and as long as the source is using the CRC module.

26.6.6 Debug Control

Name: DBGCTRL
Offset: 0x0D
Reset: 0x00
Property: RW



Bit 0 – DBGRUN Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The DMAC is halted when the CPU is halted by an external debugger.
1	The DMAC continues normal operation when the CPU is halted by an external debugger.

26.6.7 QOS Control

Name: QOSCTRL
Offset: 0x0E
Reset: 0x2A
Property: RW

Bit	7	6	5	4	3	2	1	0
			DQOS[1:0]		FQOS[1:0]		WRBQOS[1:0]	
Access								
Reset			1	0	1	0	1	0

Bits 5:4 – DQOS[1:0] Data Transfer Quality of Service

These bits define the memory priority access during the data transfer operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 3:2 – FQOS[1:0] Fetch Quality of Service

These bits define the memory priority access during the fetch operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Bits 1:0 – WRBQOS[1:0] Write-Back Quality of Service

These bits define the memory priority access during the write-back operation.

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

Value	Name	Description
0x0	DISABLE	Background (no sensitive operation)
0x1	LOW	Sensitive Bandwidth
0x2	MEDIUM	Sensitive Latency
0x3	HIGH	Critical Latency

26.6.8 Software Trigger Control

Name: SWTRIGCTRL
Offset: 0x10
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	SWTRIG[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SWTRIG[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SWTRIG[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SWTRIG[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SWTRIG[31:0] Channel x Software Trigger

26.6.9 Priority Control 0

Name: PRICTRL0
Offset: 0x14
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	RRLVLEN3					LVLPRI3[4:0]		
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RRLVLEN2					LVLPRI2[4:0]		
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RRLVLEN1					LVLPRI1[4:0]		
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RRLVLEN0					LVLPRI0[4:0]		
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 31 – RRLVLEN3 Level 3 Round-Robin Scheduling Enable

This bit controls which arbitration scheme is selected for DMA channels with priority level 3. For details on arbitration schemes, refer to .

Value	Description
0	Static arbitration scheme for channels with level 3 priority.
1	Round-robin arbitration scheme for channels with level 3 priority.

Bits 28:24 – LVLPRI3[4:0] Level 3 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN3 = 1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.

When static arbitration is enabled (RRLVLEN3 = 0) for priority level 3, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (RRLVLEN3 written to '0').

Bit 23 – RRLVLEN2 Level 2 Round-Robin Scheduling Enable

This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to .

Value	Description
0	Static arbitration scheme for channels with level 2 priority.
1	Round-robin arbitration scheme for channels with level 2 priority.

Bits 20:16 – LVLPR12[4:0] Level 2 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN2 = 1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.

When static arbitration is enabled (RRLVLEN2 = 0) for priority level 2, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (RRLVLEN2 written to '0').

Bit 15 – RRLVLEN1 Level 1 Round-Robin Scheduling Enable

For details on arbitration schemes, refer to .

Value	Description
0	Static arbitration scheme for channels with level 1 priority.
1	Round-robin arbitration scheme for channels with level 1 priority.

Bits 12:8 – LVLPR11[4:0] Level 1 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN1 = 1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.

When static arbitration is enabled (RRLVLEN1 = 0) for priority level 1, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (RRLVLEN1 written to '0').

Bit 7 – RRLVLEN0 Level 0 Round-Robin Scheduling Enable

For details on arbitration schemes, refer to .

Value	Description
0	Static arbitration scheme for channels with level 0 priority.
1	Round-robin arbitration scheme for channels with level 0 priority.

Bits 4:0 – LVLPR10[4:0] Level 0 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN0 = 1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.

When static arbitration is enabled (RRLVLEN0 = 0) for priority level 0, and the value of this bit group is non-zero, it will not affect the static priority scheme.

This bit group is not reset when round-robin arbitration gets disabled (RRLVLEN0 written to '0').

26.6.10 Priority Control 1

Name: PRICTRL1
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 26-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	RRLVLEN7			LVLPRI7[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	RRLVLEN6			LVLPRI6[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RRLVLEN5			LVLPRI5[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RRLVLEN4			LVLPRI4[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 31 – RRLVLEN7 Level 7 Round-Robin Scheduling Enable

Bits 28:24 – LVLPRI7[4:0] Level 7 Channel Priority Number

Bit 23 – RRLVLEN6 Level 6 Round-Robin Scheduling Enable

Bits 20:16 – LVLPRI6[4:0] Level 6 Channel Priority Number

Bit 15 – RRLVLEN5 Level 5 Round-Robin Scheduling Enable

Bits 12:8 – LVLPRI5[4:0] Level 5 Channel Priority Number

Bit 7 – RRLVLEN4 Level 4 Round-Robin Scheduling Enable

Bits 4:0 – LVLPRI4[4:0] Level 4 Channel Priority Number

26.6.11 Interrupt Pending

Name: INTPEND
Offset: 0x20
Reset: 0x0000
Property: RW

This register allows the user to identify the lowest DMA channel with pending interrupt.



Important: INTPEND is dedicated to channels which are multiplexed on a single interrupt line (Channels 4 to 15).

Bit	15	14	13	12	11	10	9	8
	PEND	BUSY	FERR			SUSP	TCMPL	TERR
Access	R	R	R			RW	RW	RW
Reset	0	0	0			0	0	0

Bit	7	6	5	4	3	2	1	0
				ID[4:0]				
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bit 15 – PEND Pending

This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

Bit 14 – BUSY Busy

This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

Bit 13 – FERR Fetch Error

This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

Bit 10 – SUSP Channel Suspend

This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Suspend interrupt flag.

Bit 9 – TCMPL Transfer Complete

This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Complete interrupt flag.

Bit 8 – TERR Transfer Error

This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

Bits 4:0 – ID[4:0] Channel ID

These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is

refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

26.6.12 Interrupt Status

Name: INTSTATUS
Offset: 0x24
Reset: 0x00000000
Property: R

Bit	31	30	29	28	27	26	25	24
	CHINT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHINT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHINT[31:0] Channel x Pending Interrupt

This bit is set when Channel n has a pending interrupt/the interrupt request is received.
This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.

26.6.13 Busy Channels

Name: BUSYCH
Offset: 0x28
Reset: 0x00000000
Property: R

Bit	31	30	29	28	27	26	25	24
	BUSYCH[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BUSYCH[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BUSYCH[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BUSYCH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BUSYCH[31:0] Busy Channel x

This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.

This bit is set when DMA channel n starts a DMA transfer.

26.6.14 Pending Channels

Name: PENDCH
Offset: 0x2C
Reset: 0x00000000
Property: R

Bit	31	30	29	28	27	26	25	24
	PENDCH[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PENDCH[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PENDCH[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PENDCH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PENDCH[31:0] Pending Channel x

This bit is cleared when trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to .TRIGACT.

This bit is set when a transfer is pending on DMA channel n.

26.6.15 Active Channel and Levels

Name: ACTIVE
Offset: 0x30
Reset: 0x00000000
Property: R

Bit	31	30	29	28	27	26	25	24
	BTCNT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BTCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ABUSY			ID[4:0]				
Access	R			R	R	R	R	R
Reset	0			0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	LVLEX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – BTCNT[15:0] Active Channel Block Transfer Count

These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel Active Busy flag (ABUSY) is set.

Bit 15 – ABUSY Active Channel Busy

This bit is cleared when the active transfer count is written back in the write-back memory section. This bit is set when the next descriptor transfer count is read from the write-back memory section.

Bits 12:8 – ID[4:0] Active Channel ID

These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

Bits 7:0 – LVLEX[7:0] Level x Channel Trigger Request Executing

26.6.16 Descriptor Memory Section Base Address

Name: BASEADDR
Offset: 0x34
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	BASEADDR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BASEADDR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BASEADDR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BASEADDR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BASEADDR[31:0] Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 64-bit aligned.

26.6.17 Write-Back Memory Section Base Address

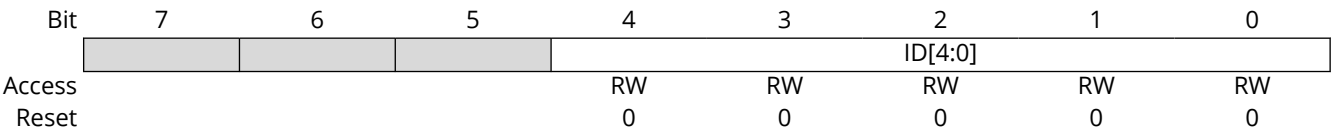
Name: WRBADDR
Offset: 0x38
Reset: 0x00000000
Property: RW

Bit	31	30	29	28	27	26	25	24
	WRBADDR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WRBADDR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WRBADDR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	WRBADDR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – WRBADDR[31:0] Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 64-bit aligned.

26.6.18 Channel ID

Name: CHID
Offset: 0x3F
Reset: 0x00
Property: RW



Bits 4:0 – ID[4:0] Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.

26.6.19 Channel Control A

Name: CHCTRLA
Offset: 0x40
Reset: 0x00
Property: RW

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		RW					RW	RW
Reset		0					0	0

Bit 6 – RUNSTDBY Channel run in standby

This bit is used to keep the DMAC channel running in standby mode.
This bit is not enable-protected.

Value	Description
0	The DMAC channel is halted in standby.
1	The DMAC channel continues to run in standby.

Bit 1 – ENABLE Channel Enable

Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
Writing a '1' to this bit will enable the DMA channel.
This bit is not enable-protected.

Value	Description
0	DMA channel is disabled.
1	DMA channel is enabled.

Bit 0 – SWRST Channel Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

26.6.20 Channel Control B

Name: CHCTRLB
Offset: 0x44
Reset: 0x00000000
Property: RW

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	31	30	29	28	27	26	25	24
							CMD[1:0]	
Access							RW	RW
Reset							0	0

Bit	23	22	21	20	19	18	17	16
	TRIGACT[1:0]							
Access	RW	RW						
Reset	0	0						

Bit	15	14	13	12	11	10	9	8
	TRIGSRC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LVL[2:0]			EVOE	EVIE	EVACT[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 25:24 – CMD[1:0] Software Command

These bits define the software commands. Refer to and .
These bits are not enable-protected.

Value	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation
0x3	-	Reserved

Value	Name	Description
0x0	NOACT	No action
0x1	SUSPEND	Channel suspend operation
0x2	RESUME	Channel resume operation

Bits 23:22 – TRIGACT[1:0] Trigger Action

These bits define the trigger action used for a transfer.

Value	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x1	-	Reserved
0x2	BEAT	One trigger required for each beat transfer
0x3	TRANSACTION	One trigger required for each transaction

Value	Name	Description
0x0	BLOCK	One trigger required for each block transfer
0x2	BEAT	One trigger required for each beat transfer

Value	Name	Description
0x3	TRANSACTION	One trigger required for each transaction

Bits 15:8 – TRIGSRC[7:0] Trigger Source

These bits define the peripheral trigger which is source of the transfer. For additional information on trigger selection and trigger modes, refer to and CHCTRLB.TRIGACT.
Refer to the table for finding out the value corresponding to each module trigger source.



Important: Refer to the for the list of peripherals and peripheral instances present in each variant.

Value	Name	Description
	PERIPHERAL	Peripheral trigger n selection
0x00	DISABLE	Only software/event triggers

Bits 7:5 – LVL[2:0] Channel Arbitration Level

These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For additional information on arbitration schemes, refer to the .
These bits are not enable-protected.

Value	Name	Description
0x0	LVL0	Channel Priority Level 0
0x1	LVL1	Channel Priority Level 1
0x2	LVL2	Channel Priority Level 2
0x3	LVL3	Channel Priority Level 3

Bit 4 – EVOE Channel Event Output Enable

This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the descriptor Event Output Selection (.EVOSEL).
This bit is available only for the 4 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event generation is disabled.
1	Channel event generation is enabled.

Bit 3 – EVIE Channel Event Input Enable

This bit is available only for the 4 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Description
0	Channel event action will not be executed on any incoming event.
1	Channel event action will be executed on any incoming event.

Bits 2:0 – EVACT[2:0] Event Input Action

These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in the CHCTRLB register of the channel is set.
These bits are available only for the 4 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

Value	Name	Description
0x0	NOACT	No action
0x1	TRIG	Normal Transfer and Conditional Transfer on Strobe trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer

EVACT (continued)

Value	Name	Description
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action
0x7	-	Reserved

Value	Name	Description
0x0	NOACT	No action
0x1	TRIG	Transfer and periodic transfer trigger
0x2	CTRIG	Conditional transfer trigger
0x3	CBLOCK	Conditional block transfer
0x4	SUSPEND	Channel suspend operation
0x5	RESUME	Channel resume operation
0x6	SSKIP	Skip next block suspend action

26.6.21 Channel Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x4C
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						RW	RW	RW
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

26.6.22 Channel Interrupt Enable Set

Name: CHINTENSET
Offset: 0x4D
Reset: 0x00
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						RW	RW	RW
Reset						0	0	0

Bit 2 – SUSP Channel Suspend Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

Value	Description
0	The Channel Suspend interrupt is disabled.
1	The Channel Suspend interrupt is enabled.

Bit 1 – TCMPL Channel Transfer Complete Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

Value	Description
0	The Channel Transfer Complete interrupt is disabled.
1	The Channel Transfer Complete interrupt is enabled.

Bit 0 – TERR Channel Transfer Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

Value	Description
0	The Channel Transfer Error interrupt is disabled.
1	The Channel Transfer Error interrupt is enabled.

26.6.23 Channel Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x4E
Reset: 0x00
Property: RW

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						SUSP	TCMPL	TERR
Access						RW	RW	RW
Reset						0	0	0

Bit 2 – SUSP Channel Suspend

This flag is cleared by writing a '1' to it.

This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.

For details on available software commands, refer to .CMD.

For details on available event input actions, refer to .EVACT.

For details on available block actions, refer to .BLOCKACT.

Bit 1 – TCMPL Channel Transfer Complete

This flag is cleared by writing a '1' to it.

This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

Bit 0 – TERR Channel Transfer Error

This flag is cleared by writing a '1' to it.

This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.

26.6.24 Channel Status

Name: CHSTATUS
Offset: 0x4F
Reset: 0x00
Property: R

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

Bit	7	6	5	4	3	2	1	0
						FERR	BUSY	PEND
Access						R	R	R
Reset						0	0	0

Bit 2 – FERR Channel Fetch Error
This bit is cleared when a software resume command is executed.
This bit is set when an invalid descriptor is fetched.

Bit 1 – BUSY Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.
This bit is set when the DMA channel starts a DMA transfer.

Bit 0 – PEND Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, refer to .TRIGACT.
This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.

26.7 DMA Descriptor Register Summary

26.7.1 Block Transfer Control

Name: BTCTRL
Offset: 0x00
Reset: 0x0000
Property: RW

Table 26-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	STEPSIZE[2:0]			STEPSEL	DSTINC	SRCINC	BEATSIZE[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				BLOCKACT[1:0]		EVOSEL[1:0]		VALID
Access								
Reset				0	0	0	0	0

Bits 15:13 – STEPSIZE[2:0] Address Increment Step Size

Value	Name	Description
0x0	X1	Next ADDR = ADDR + (BEATSIZE+1) * 1
0x1	X2	Next ADDR = ADDR + (BEATSIZE+1) * 2
0x2	X4	Next ADDR = ADDR + (BEATSIZE+1) * 4
0x3	X8	Next ADDR = ADDR + (BEATSIZE+1) * 8
0x4	X16	Next ADDR = ADDR + (BEATSIZE+1) * 16
0x5	X32	Next ADDR = ADDR + (BEATSIZE+1) * 32
0x6	X64	Next ADDR = ADDR + (BEATSIZE+1) * 64
0x7	X128	Next ADDR = ADDR + (BEATSIZE+1) * 128

Bit 12 – STEPSEL Step Selection

Value	Name	Description
0x0	DST	Step size settings apply to the destination address
0x1	SRC	Step size settings apply to the source address

Bit 11 – DSTINC Destination Address Increment Enable

Bit 10 – SRCINC Source Address Increment Enable

Bits 9:8 – BEATSIZE[1:0] Beat Size

Value	Name	Description
0x0	BYTE	8-bit bus transfer
0x1	HWORD	16-bit bus transfer
0x2	WORD	32-bit bus transfer

Bits 4:3 – BLOCKACT[1:0] Block Action

Value	Name	Description
0x0	NOACT	Channel will be disabled if it is the last block transfer in the transaction
0x1	INT	Channel will be disabled if it is the last block transfer in the transaction and block interrupt
0x2	SUSPEND	Channel suspend operation is completed
0x3	BOTH	Both channel suspend operation and block interrupt

Bits 2:1 – EVOSEL[1:0] Event Output Selection

Value	Name	Description
0x0	DISABLE	Event generation disabled
0x1	BLOCK	Event strobe when block transfer complete
0x3	BEAT	Event strobe when beat transfer complete

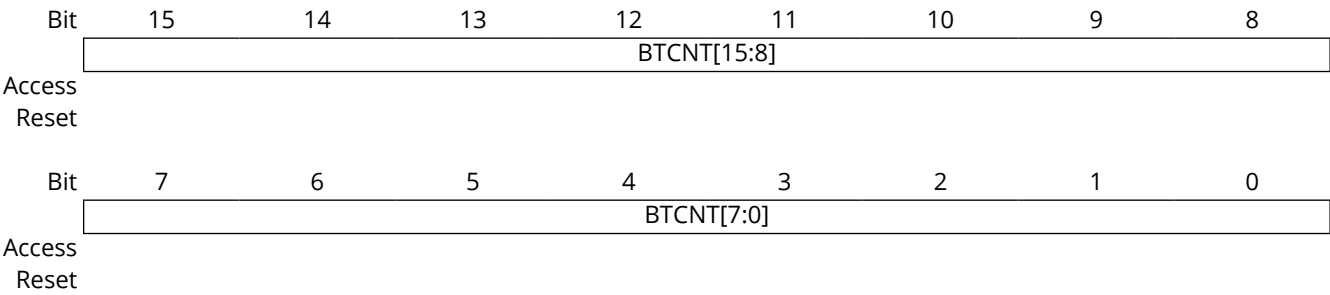
Bit 0 – VALID Descriptor Valid

26.7.2 Block Transfer Count

Name: BTCNT
Offset: 0x02
Reset: None
Property: RW

Table 26-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—



Bits 15:0 – BTCNT[15:0] Block Transfer Count

26.7.3 Block Transfer Source Address

Name: SRCADDR
Offset: 0x04
Reset: None
Property: RW

Table 26-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access								
Reset								

Bits 31:0 – SRCADDR[31:0] Transfer Source Address

26.7.4 Block Transfer Destination Address

Name: DSTADDR
Offset: 0x08
Reset: None
Property: RW

Table 26-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DSTADDR[31:24]							
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	DSTADDR[23:16]							
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	DSTADDR[15:8]							
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	DSTADDR[7:0]							
Access								
Reset								

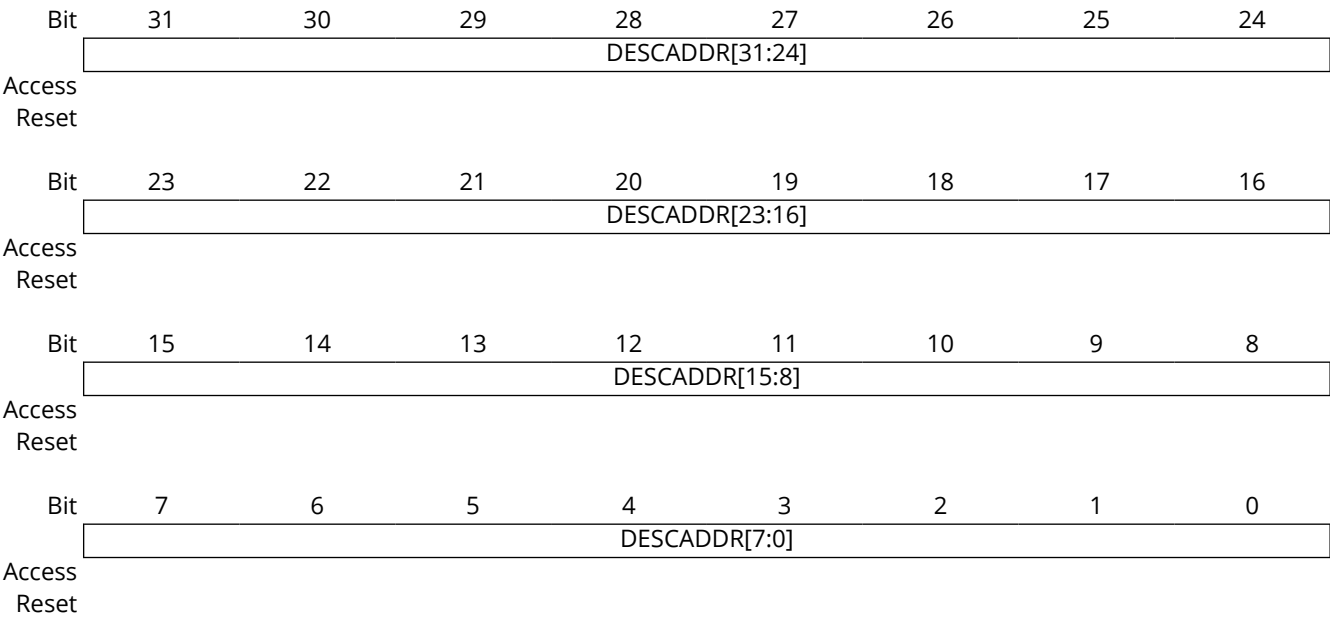
Bits 31:0 – DSTADDR[31:0] Transfer Destination Address

26.7.5 Next Descriptor Address

Name: DESCADDR
Offset: 0x0C
Reset: None
Property: RW

Table 26-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—



Bits 31:0 – DESCADDR[31:0] Next Descriptor Address

27. Supply Controller (SUPC)

27.1 Overview

The Supply Controller (SUPC) manages the voltage reference, power supply and supply monitoring of the device. It is also able to control multiple output pins named "SUPC_OUT[1:0]".

The SUPC controls all the voltage regulators supplying the multiple 1.2v domain: switchable core domain (PD_CORE_SW), RAM power domain (PD_CORE_RAMx), backup (PD_CORE_BU) domain . It sets the voltage regulators according to the sleep modes, or the user configuration. It also controls a User LDO regulator to power an external component.

The SUPC controls several Brown-Out Reset (BOR).BOR_VDDREG monitors the voltage applied to the regulators (VDDREG), BOR_VDDIO monitors the voltage applied to IOs. In backup mode, a dedicated Duty Cycle BOR (DCBOR) is able to monitor both VDDREG and VDDIO/AVDD. BOR12 tightly coupled with the capless regulator guarantees the internal voltage of the core (VDDCORE).

The SUPC controls a configurable Low-Voltage Detector used to monitor VDDREG in both directions.

The SUPC controls an internal reference voltage and a current dependent on the temperature which can be used by analog modules like the ADC.

The SUPC also controls the device's charge pumps.

27.2 Features

The SUPC controls the following analog supply elements:

- Voltage Regulator System
 - Capless voltage regulators called VREGSW is used in active, idle and standby mode, to supply PD_CORE_SW power domain.
 - Capless voltage regulator called VREGRAM used in active, idle, standby and hibernate mode to supply VDDCORE_RAM domain and PLL
- Voltage Reference System (Bandgap)
 - Reference voltage for ADC
 - Temperature sensor
 - Low power mode in standby sleep mode
 - Charge Pump for I/O pad and analog cells as PTC/AC/ADC in case of low VDD voltage
- 3.3V Brown-out Reset (BOR) Detector
 - Three instances of BOR are used when calibrated to monitor VDDIO, AVDD and VDDREG power supply voltages, during Power Up, Active mode and Standby Sleep mode
 - Programmable threshold value loaded from USER CFG page at startup
 - Triggers resets
 - Operating modes:Continuous mode and Low power mode in standby sleep mode.
- 3.3V Low-Power Brown-out Reset (DCBOR)
 - Threshold values loaded from USER CFG page of CFM Flash Memory
 - Triggers resets
 - Operating modes: Continuous mode and Sampled mode (with programmable sampling frequency)
- 1.2V Brown-Out Reset (BOR12)
 - Monitors VDDCORE power supply voltage
 - Tightly coupled with the capless regulator

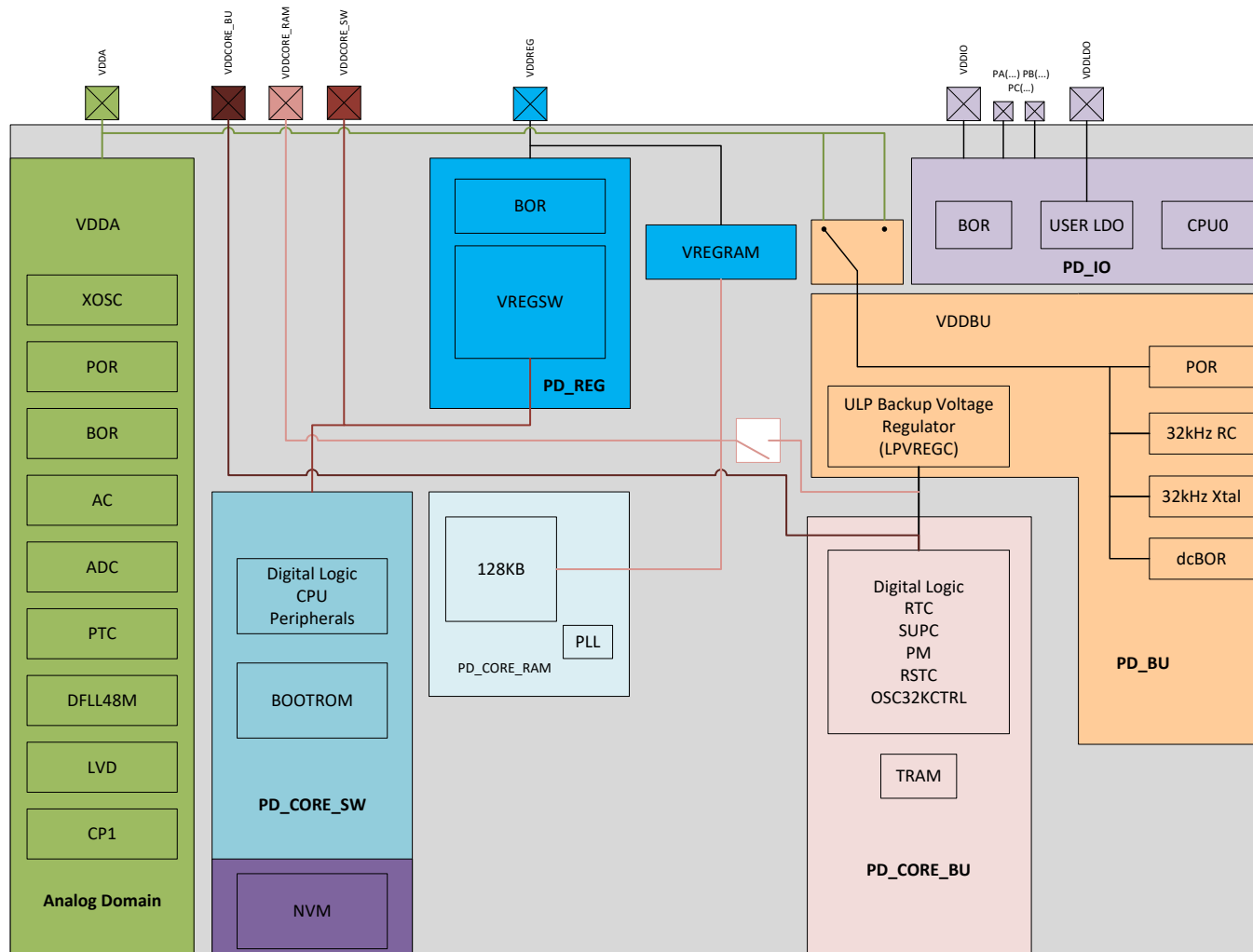
- Triggers resets
- 3.3V Programmable Low-voltage Detector
 - Monitors VDDREG
 - Configurable threshold and direction
 - Can trigger Interrupt
- Output pins
 - Pin toggling on RTC event or by SUPC in Backup mode

27.3 Block Diagram

As shown in the following figure, there are several power supply pins:

- AVDD powers the Backup power domain as well as the ADCs, Analog Comparator (AC), and the Peripheral Touch Controller (PTC)
- VDDREG powers the internal regulators for the VDDCORE_SW, VDDCORE_RAM, and VDDCORE_PLL power domains
- VDDIO powers I/O lines, an External Crystal Oscillator (XOSC), the 48 MHz Digital Frequency Locked Loop (DFLL48), and three charge pumps which support the Analog Comparator (AC) and the ADCs on the device (ADC0-3 and PTC ADC). Most VDDIO pins also power the device's Flash panel via double bonding.
- VDDUSB3V0 and VDDUSB3V1 power the USB ports on the device

Figure 27-1. Supply Controller Block Diagram



Power domains (PDs) shown above are not independent. VDDCORE_*, VDDIO, and VDDREG share the same ground, GND. But AVDD has its own ground, AVSS. AVDD and VDDIO inputs must share the same supply, VDD.

The block diagram above also shows a variety of internally regulated power domains (VDDCORE PDs), nominally at 1.2V:

- **VDDCORE_BU:** Powers the backup domain. it contains peripherals that remain powered in the Backup Sleep mode
- **VDDCORE_SW:** The main voltage domain for the CPU, bus, and most peripherals, which can be switch off (hence the _SW suffix)
- **VDD CORE RAM:** This domain is used to retain the devices SRAM

The device's internal main voltage regulators have three different modes, controlled by the Supply Controller (SUPC):

- **Active Run Mode:** The default mode when the CPU and peripherals are running
- **Idle/Standby:** When the CPU and peripherals are in standby
- **Hibernate/Backup/Off Mode:** When the chip is in backup mode, the internal regulator is off, the VDDCORE_SW core power domain is OFF. The VDDCORE_BU backup domain is powered by the backup regulator (Low Power Voltage Regulator for Core - LPVREGC).

27.4 Signals Description

Table 27-1. Signals Description

Signal Name	Type	Description
SUPC_OUT0	Digital Output	SUPC Output 0
SUPC_OUT1	Digital Output	SUPC Output 1

27.5 Functional Description

27.5.1 Voltage Regulators

27.5.1.1 Enabling, Disabling, and Resetting

The main regulators output supply level is automatically defined by the Sleep mode selected in the Power Manager module.

27.5.1.2 VDDCORE Control

The VDDCORE supplies (VDDCORE_SW, VDDCORE_RAM, VDDCORE_BU, VDDCORE_USB if enabled in Standby mode).

27.5.1.3 Charge Pump for Low VDDIO Voltage

This is highly recommended to activate charge pumps when VDDA/VDDIO is too low. The control of Charge pump n ($n = 0,1$) is managed by setting the corresponding bit in VREGCTRL.CPEN[1:0]. When CPEN[n] bit is set, the enable and auto-enable bits of the charge pump[N] are set. In standby mode, the charge pumps are automatically turned OFF except if a consumer (PTC, ADC or AC) of the charge pump is requesting the charge pump.

27.5.2 Voltage References

The reference voltages are generated by a functional analog block controlled by the SUPC. It is providing a fixed-voltage source VREF for modules as ADC and AC as well as a temperature dependent voltage in support of a temperature sensor.

27.5.2.1 Initialization

After reset, the voltage reference output is enabled and running at full power mode, and the temperature sensor is disabled.

27.5.2.2 Enabling, Disabling, and Resetting

The temperature sensor is enabled/disabled by setting/clearing the Temperature Sensor Enable bit in the Voltage Reference register (VREFCTRL.TSEN).

27.5.3 Output and Input Pins

The SUPC can drive two outputs. By writing a '1' to the corresponding Output Enable bit in the Backup Output Control register (BKOUT.EN), the SUPC_OUT n pin is driven by the SUPC.

The OUT pin can be set by writing a '1' to the corresponding Set Output bit in the Backup Output Control register (BKOUT.SETx). The OUT pin can be cleared by writing a '1' to the corresponding CLR bit (BKOUT.CLRx).

If the "Toggle Output Mode X is set to "RTCTGL", the corresponding SUPC_OUT n pin will toggle when an RTC event occurs. Note that before configuring this mode, this is recommended to configure the RTC module properly. If the "Toggle Output Mode X is set to "BKUPTGL", the corresponding SUPC_OUT n pin will toggle to one when the device goes to Backup Mode. It will toggle to zero when exiting Backup Mode.

The BKIN register reflects the current state of the of the corresponding BKOUT pin if enabled (SUPC.BKOUT.EN = 1).

27.5.4 Brown-out Reset (BOR)

27.5.4.1 Initialization

- BOR_VDDREG: After power-up or user reset, the BOR monitoring VDDREG is active. Its hysteresis values are loaded from NVM calibration User Config (UCFG) and stored in the BRCFGUSMOR_BOR register.
- BOR_VDDIO: After power-up or user reset, the BOR monitoring VDDIO is active. Its threshold and hysteresis values are loaded from NVM User Config (UCFG) and stored in the BRCFGUSMOR_BOR register.
- BOR_VDDA: After power-up or user reset, the BOR monitoring VDDA is active. Its threshold and hysteresis values are loaded from NVM Calibration User Config (UCFG) and stored in the BRCFGUSMOR_BOR register.
- DCBOR: After power-up or user reset, the DCBOR used in backup mode is inactive. Its threshold and hysteresis values are loaded from NVM Calibration User Config (UCFG) and stored in the BRCFGUDSSMOR_BOR register.

27.5.4.2 3.3V Regular Brown-out Reset (VDDREG_BOR or VDDIO/VDDA_BOR)

In all modes except the Backup mode the BOR_VDDREG and BOR_VDDIO/VDDA compare respectively the VDDREG and VDDIO/VDDA voltage with the VDDREG and VDDIO brown-out threshold level. This level is set during power-up when NVM calibration values are loaded. When VDDREG or VDDIO crosses below the brown-out threshold level, BOR generates a Reset.

27.5.4.3 3.3V Duty Cycle Brown-Out Reset (DCBOR)

To reduce power consumption, DCBOR is used in sampling mode. The Sampling Mode is a low-power mode where the DCBOR is being repeatedly enabled on a sampling clock's ticks. The DCBOR will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick. The frequency of the clock ticks is controlled by the Prescaler Select bit groups in the BOR register (BOR.DCBORPSEL). Please refer to "BOR – BOR Control" SFR for more details.

27.5.4.4 BOR Filtering

For BOR reset generation we have an input filter to filter-out any glitches which are less than 100nS. Supplementary BOR filtering is also possible for BOR_VDDA, BOR_VDDIO, and BOR_VDDREG by using the BOR.BORFILT bitfield.

27.5.5 Programmable Low Voltage Detector

The programmable LVD is able to monitor the VDDIO input power supply.

Before enabling the LVD by setting LVD.ENABLE bit, the threshold level bits field (LVD.LEVEL) and the direction (LVD.DIR) have to be configured. The LVD register is Enable-Protected, meaning that it can only be written when the LVD is disabled (LVD.ENABLE). As long as the Enable bits is '1', any writes to LVD register will be discarded and a bus error will be generated. The Enable bits are not Enable-Protected.

27.5.5.1 Settling Time

After being enabled, a start-up delay is required before the result of the comparison is ready. During the startup-time, the output value is zero (STATUS.LVDET).

After the start-up time has passed, the LVDET ready bit is set (STATUS.LVDRDY) indicating that STATUS.LVDET bit is available, and the appropriate LVD interrupt flag or output event are also generated. New comparisons are performed continuously until LVD.ENABLE bit is written to zero. The start-up time applies only to the first comparison. If LVD is disabled (LVD.ENABLE is written to 0), the STATUS.LVDRDY bit will take time to fall because of resynchronization. As a consequence, this is forbidden to re-enable immediately the LVD if the STATUS.LVDRDY field is still one.

27.5.5.2 LVD Interrupt and Event

- **LVD Output Interrupt source:** LVD Output interrupt source status (STATUS.LVDET) is set to one when the VDDIO voltage value crosses the programmed threshold level (LVD.LEVEL) in the right direction (LVD.DIR). It is reset to zero when the VDDIO voltage value crosses the threshold level in the wrong direction (LVD.DIR). Note that if the LVD is just enabled and the voltage is already below the threshold (if DIR=1) or the voltage is already above the threshold (if DIR=0), then the LVD output interrupt is also generated.

- **LVD Output event:** The LVD can generate output event when the VDDIO voltage value crosses the programmed threshold level (LVD.LEVEL) in the right direction (LVD.DIR). To enable this feature, the LVD.OEVEN bit has to be set to one. Refer to the EVSYS Event System for details on configuring the event system.

27.5.5.3 LVD Behavior in Sleep Mode

LVD is off in hibernate and backup sleep modes.

LVD is automatically turned OFF in standby mode except if the LVD.RUNSTDBY bit is set.

27.5.6 Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- LVD Ready (LVDRDY), synchronous
- LVD Detection (LVDDET), asynchronous

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled via INTENSET. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The SUPC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

27.5.7 Power Management

The SUPC can operate in all sleep modes except backup sleep mode.

27.5.8 Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

27.5.9 Register Access Protection

Registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC). Here is the list of the write-protected registers:

- INTENCLR
- INTENSET
- BOR
- LVD
- VREGCTRL

- VREFCTRL
- BBPS
- BKOUT

27.5.10 Temperature Calibration

27.5.10.1 Summary

With calibration, it is possible to achieve +/-5% performance from the onboard bandgap temperature sensor. This measurement can provide important information about the die temperature that may be critical in an operating environment with elevated ambient temperatures. Refer to the Electrical Characteristics section of the data sheet for maximum die temperature. Refer to the *Power and Temperature Considerations* section.

The following table is a summary of calibration fuses and mathematical representations. The values provided in these fuses result from a calibration procedure that is included in the final test program for each individual part. Some of these values are unique for each individual part which necessitates the implementation of the mathematical routines described below and more comprehensively in the document mentioned above.

Table 27-2.

OTP Cal Fuse Name	Address	Bitfield Name	Bitmask	Default	Description
FUSES_FCCFG69_REG	0x0A007194	ROOM_ADC_VAL_PTAT	0x00000fff	992	12-bit ADC conversion at room temperature of PTAT in LSB
-	-	ROOM_ADC_VAL_BUFF1V2	0x00fff000	1489	12-bit ADC conversion at room temperature of Buffered 1.2V in LSB
-	-	ROOM_TEMP_VAL_INT	0xff000000	25	Integer part of room temperature in °C
FUSES_FCCFG70_REG	0x0A007198	ROOM_TEMP_VAL_DEC	0x0000000f	0	Decimal part of room temperature in 0.1°C
-	-	ROOM_VREF_VAL	0x0000ff0	3300	voltage in mV of VREF = VDDANA
-	-	HOT_TEMP_VAL_DEC	0x000f0000	0	Decimal part of hot temperature in 0.1°C
-	-	Reserved	0xffff0000	-	-
FUSES_FCCFG71_REG	0x0A00719C	HOT_ADC_VAL_PTAT	0x00000fff	1325	12-bit ADC conversion at hot temperature in LSB of PTAT in LSB
-	-	HOT_ADC_VAL_BUFF1V2	0x00fff000	1489	12-bit ADC conversion at hot temperature of Buffered 1.2V in LSB
-	-	HOT_TEMP_VAL_INT	0xff000000	125	Integer part of hot temperature in °C

27.5.10.2 Slope Calculation

T_{room} (°C): Floating point combination of calibrations values ROOM_TEMP_INT and ROOM_TEMP_DEC representing the ambient temperature calibration point.

T_{hot} (°C): Floating point combination of calibrations values HOT_TEMP_INT and HOT_TEMP_DEC representing the hot temperature calibration point.

Δ_{ptat-cal} (bit): Raw bit value difference of the calibration value HOT_ADC_VAL_PTAT - ROOM_ADC_VAL_PTAT.

V_{ref-cal} (mV): Calibration reference voltage ROOM_VREF_VAL in mV.

$\frac{dV}{dT} \left(\frac{V}{^{\circ}C} \right)$ – *slope*: Change of voltage with respect to temperature.

$$\frac{dV}{dT} \left(\frac{V}{^{\circ}C} \right) = \frac{V_{ref-cal}}{(1000)(4095)} \left(\frac{V}{bit} \right) \cdot \frac{\Delta_{ptat-cal}}{T_{hot} - T_{room}} \left(\frac{bit}{^{\circ}C} \right)$$

27.5.10.3 Die Temperature Calculation

N_{adc} (bit): Raw bit value of current ADC read.

$\Delta_{ptat-app}$ (bit): Raw bit value difference between HOT_ADC_VAL_PTAT (Vref ratio compensated) and N_{ADC} .

$$\Delta_{ptat-app} (bit) = HOT\_ADC\_VAL\_PTAT - N_{adc} * (V_{ref-app} / V_{ref-cal})$$

V_{ht-cal} (V): Voltage representation of hot temperature calibration value.

$$\left(V\right) = \frac{dV}{dT} \left(\frac{V}{^{\circ}C}\right) \cdot T_{hot} (^{\circ}C)$$

$V_{ref-app}$ (mV): Application reference voltage in mV

$V_{diff-app}$ (V): Voltage representation of the raw bit value difference between the hot temperature calibration reference and the current ADC read value (compensated for the V_{ref} ratio already happened, hence $V_{ref-cal}$ is used).

$$V_{diff-app} (V) = \frac{V_{ref-cal}}{(1000)(4095)} \left(\frac{V}{bit}\right) \cdot \Delta_{ptat-app} (bit)$$

T_j : Junction (die) temperature.

$$T_j (^{\circ}C) = \frac{(V_{ht-cal} - V_{diff-app})}{\left(\frac{dV}{dT}\right)}$$

27.6 Peripheral Dependencies

Table 27-3. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
SUPC	0x45002000	3 : LVDDDET/ LVDRDY/ ULDORDY/ ULDOOVHEAT	APB: CLKMSK[0] MASK4	---	4 STATUS0 PERID4		<u>Generators:</u> LVDET: CHANNEL[1]	PD_CORE_SW

27.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	INTENCLR	31:24								
		23:16								
		15:8					ADDVREGRDY[3:0]			
		7:0	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEA T	ULDORDY	LVDRDY	LVDET
0x04	INTENSET	31:24								
		23:16								
		15:8					ADDVREGRDY[3:0]			
		7:0	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEA T	ULDORDY	LVDRDY	LVDET
0x08	INTFLAG	31:24								
		23:16								
		15:8					ADDVREGRDY[3:0]			
		7:0	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEA T	ULDORDY	LVDRDY	LVDET
0x0C	STATUS	31:24								
		23:16								
		15:8					ADDVREGRDY[3:0]			
		7:0	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEA T	ULDORDY	LVDRDY	LVDET
0x10	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0								BOR
0x14	BOR	31:24								
		23:16								
		15:8							BORFILT[1:0]	
		7:0		DCBORPSEL[2:0]						ACTION
0x18	LVD	31:24								
		23:16					LEVEL[3:0]			
		15:8								
		7:0				RUNSTDBY	OEVEN	DIR	ENABLE	
0x1C	VREGCTRL	31:24	BKUP_VLD	SRAM_VLD			AVREGSTDBY[3:0]			
		23:16					AVREGEN[3:0]			
		15:8	ULDOLEVEL[1:0]		ULDOSTDBY	ULDOEN	CPEN[3:0]			
		7:0	SCAPEN		LVHIB	LVSTDBY		OFFSTDBY	VREGOUT[1:0]	
0x20	VREFCTRL	31:24								
		23:16								
		15:8								
		7:0				TSEN			LPHIB	LPSTDBY
0x24	BBPS	31:24								
		23:16								
		15:8								
		7:0			SELVBATMON	ENVBATMON		WAKEEN	WKONVBATE N	CONF
0x28	BKOUT	31:24							TGLOM[1:0]	
		23:16								SET
		15:8								CLR
		7:0								EN
0x2C	BKIN	31:24								
		23:16								
		15:8								
		7:0					BKIN[3:0]			

27.7.1 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 27-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						ADDVREGRDY[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEAT	ULDORDY	LVDRDY	LVDET
Access	R/W	R/W	R/W	R/W	R/W	R/W	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – ADDVREGRDY[3:0] Additional Regulator ready x Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to the bit disables the ADDVREGRDY interrupt.

Each bit will read as the current value of the ADDVREGRDY interrupt enable.

Bits 7:5 – BORVDDUSB[2:0] BORVDDUSB Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to the bit disables the BORVDDUSB interrupt.

Each bit will read as the current value of the BORVDDUSB interrupt enable.

Bit 4 – BORVDDIOB BORVDDIOB Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit disables the BORVDDIOB interrupt.

This bit will read as the current value of the BORVDDIOB interrupt enable.

Bit 3 – ULDOOVHEAT User LDO Regulator OverHeat Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit disables the ULDOOVHEAT interrupt.

This bit will read as the current value of the ULDOOVHEAT interrupt enable.

Bit 2 – ULDORDY User LDO Regulator Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit disables the ULDORDY interrupt.

This bit will read as the current value of the ULDORDY interrupt enable.

Bit 1 – LVDRDY Low Voltage Detector Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit disables the LVDRDY interrupt.

This bit will read as the current value of the LVDRDY interrupt enable.

Bit 0 – LVDET Low Voltage Detector Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit disables the LVDET interrupt.

This bit will read as the current value of the LVDET interrupt enable.

27.7.2 Interrupt Enable Set

Name: INTENSET
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 27-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						ADDVREGRDY[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEAT	ULDORDY	LVDRDY	LVDET
Access	R/W	R/W	R/W	R/W	R/W	R/W	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – ADDVREGRDY[3:0] Additional Regulator ready x Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to the bit enables the ADDVREGRDY interrupt.

Each bit will read as the current value of the ADDVREGRDY interrupt enable.

Bits 7:5 – BORVDDUSB[2:0] BORVDDUSB Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to the bit enables the corresponding BORVDDUSB interrupt.

Each bit will read as the current value of the BORVDDUSB interrupt enable.

Bit 4 – BORVDDIOB BORVDDIOB Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit enables the BORVDDIOB interrupt.

This bit will read as the current value of the BORVDDIOB interrupt enable.

Bit 3 – ULDOOVHEAT User LDO Regulator OverHeat Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit enables the ULDOOVHEAT interrupt.

This bit will read as the current value of the ULDOOVHEAT interrupt enable.

Bit 2 – ULDORDY User LDO Regulator Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit enables the ULDORDY interrupt.

This bit will read as the current value of the ULDORDY interrupt enable.

Bit 1 – LVDRDY Low Voltage Detector Ready Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit enables the LVDRDY interrupt.

This bit will read as the current value of the LVDRDY interrupt enable.

Bit 0 – LVDET Low Voltage Detector Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit enables the LVDET interrupt.

This bit will read as the current value of the LVDET interrupt enable.

27.7.3 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x008
Reset: 0x00000000
Property: RW

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 27-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						ADDVREGRDY[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEAT	ULDORDY	LVDRDY	LVDET
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – ADDVREGRDY[3:0] Additional Regulator ready x Interrupt. Set to one if additional regulator is ready meaning that output voltage is correct.

Set by hardware when the Additional Regulator is ready and the output voltage is correct.

Write one to clear the bit.

Bits 7:5 – BORVDDUSB[2:0] BORVDDUSB Interrupt. Set to one if VDDUSB issue is detected.

Set by hardware when a Brown-Out has been detected on the corresponding VDD_USB.

Write one to clear the corresponding bit.

Bit 4 – BORVDDIOB BORVDDIOB Interrupt. Set to one if VDDIOB issue is detected.

Set by hardware when a Brown-Out has been detected for VDDIOB.

Write one to clear the corresponding bit.

Bit 3 – ULDOOVHEAT User LDO Regulator OverHeat Interrupt. Is set to one if overheat condition is detected.

Set by hardware when the User LDO overheat condition is detected.

Write one to clear the corresponding bit.

Bit 2 – ULDORDY User LDO Regulator Ready Interrupt. Set to one if ULDO is ready meaning that output voltage is correct.

Set by hardware when the User LDO is ready to operate.

Write one to clear the corresponding bit.

Bit 1 – LVDRDY Low Voltage Detector Ready Interrupt. Set to one if LVD is ready to operate.

Set by hardware the Low Voltage Detector is ready to operate.

Write one to clear the corresponding bit.

Bit 0 – LVDET Low Voltage Detector Interrupt. Set to one if VDDIO crosses the threshold voltage in the good direction according to LVD.DIR.

Set by hardware if VDDIO crosses the threshold voltage in a “good” direction according to LVD.DIR.

Write one to clear the corresponding bit.

27.7.4 Flag status

Name: STATUS
Offset: 0x00C
Reset: 0x00000000
Property: R

Table 27-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					ADDVREGRDY[3:0]			
Access					R	R	R	R
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BORVDDUSB[2:0]			BORVDDIOB	ULDOOVHEAT	ULDORDY	LVDRDY	LVDET
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – ADDVREGRDY[3:0] Additional Regulator ready x Status. One if voltage is OK. It corresponds to vreg_ready_mv signal of additional regulator.

Set by hardware when the voltage corresponding Additional Regulator is ok.

Bits 7:5 – BORVDDUSB[2:0] BORVDDUSB Status. One if VDDUSB is OK. It corresponds to bor_vddusb_n_mv signal of SMOR.

Set by hardware when VDD_USB is OK.

Bit 4 – BORVDDIOB BORVDDIOB Status. One if VDDIOB is OK. It corresponds to bor_vddiob_mv signal of SMOR.

Set by hardware when a Brown-Out has been detected for VDDIOB.

Bit 3 – ULDOOVHEAT User LDO Regulator OverHeat Status. It corresponds to vreg_overheat_event_mv signal from ULDO.

Set by hardware when the User LDO overheat condition is detected.

Bit 2 – ULDORDY User LDO regulator Status. It corresponds to vreg_ready_mv signal from ULDO.

Set by hardware when the User LDO is ready to operate.

Bit 1 – LVDRDY Low Voltage Detector Ready Status

Set by hardware when the Low Voltage Detector is ready.

Bit 0 – LVDET Low Voltage Detector Status.

Set to one if VDDIO crosses the threshold voltage in the “good” direction according to LVD.DIR.

27.7.5 Synchronisation Busy

Name: SYNCBUSY
Offset: 0x010
Reset: 0x00000000
Property: R

Table 27-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								BOR
Access								R
Reset								0

Bit 0 – BOR BOR Synchronization Busy

When hardware sets this bit, no writes are permitted to the BOR register. After writing to the BOR register wait for this flag to clear before making additional writes.

27.7.6 BOR Control

Name: BOR
Offset: 0x014
Reset: 0x00000000
Property: RW

Table 27-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							BORFILT[1:0]	
Access							RW	RW
Reset							0	0
Bit	7	6	5	4	3	2	1	0
		DCBORPSEL[2:0]						ACTION
Access		RW	RW	RW				RW
Reset		0	0	0				0

Bits 9:8 – BORFILT[1:0] BOR filtering

Value	Name	Description
0x0	NOFILT	No digital filtering
0x1	FILT32US	32us filtering
0x2	FILT125US	125us filtering
0x3	FILT250US	250us filtering

Bits 6:4 – DCBORPSEL[2:0] Duty Cycle BOR Prescaler Select

Value	Name	Description
0x0	NODIV	Not Divided
0x1	DIV2	Divide clock by 2
0x2	DIV4	Divide clock by 4
0x3	DIV8	Divide clock by 8
0x4	DIV16	Divide clock by 16
0x5	DIV32	Divide clock by 32
0x6	DIV64	Divide clock by 64
0x7	DIV128	Divide clock by 128

Bit 0 – ACTION Action when Threshold Crossed

Value	Name	Description
0x0	RESET	The BOR generates a reset
0x1	BBKUP	The BOR puts the device in battery backup sleep mode

27.7.7 LVD Control

Name: LVD
Offset: 0x018
Reset: 0x00000000
Property: RW

Table 27-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
					LEVEL[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				RUNSTDBY	OEVEN	DIR	ENABLE	
Access				RW	RW	RW	RW	
Reset				0	0	0	0	

Bits 19:16 – LEVEL[3:0] Threshold Level.

Value	Description
1111	3.630 V
1110	3.330 V
1101	3.000 V
1100	2.927 V
1011	2.824 V
1010	2.727 V
1001	2.500 V
1000	2.400 V
0111	2.308 V
0110	2.243 V
0101	2.202 V
0100	2.124 V
0011	2.017 V
0010	1.890 V
0001	1.805 V
0000	1.727 V

Bit 4 – RUNSTDBY Run during Standby

Value	Description
0	Run during standby disabled
1	Run during standby enabled

Bit 3 – OEVEN Output Event Enable

Value	Description
0	Output events disabled
1	Output events enabled

Bit 2 – DIR Direction

Value	Name	Description
0x0	rising	rising detection
0x1	falling	falling detection

Bit 1 – ENABLE Enable

Value	Description
0	Low Voltage Detection disabled
1	Low Voltage Detection enabled

27.7.8 VREG Control

Name: VREGCTRL
Offset: 0x01C
Reset: 0x00000004
Property: RW

Note: During normal operation, all voltage regulators that are in use must be left in the On state to allow for the proper transition between different low-power/standby states.

Table 27-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BKUP_VLD	SRAM_VLD				AVREGSTDBY[3:0]		
Access	RW	RW			RW	RW	RW	RW
Reset	0	0			0	0	0	0
Bit	23	22	21	20	19	18	17	16
						AVREGEN[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ULDOLEVEL[1:0]		ULDOSTDBY	ULDOEN		CPEN[3:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SCAPEN		LVHIB	LVSTDBY		OFFSTDBY	VREGOUT[1:0]	
Access	R/W		RW	RW		RW	RW	RW
Reset	0		0	0		1	0	0

Bit 31 – BKUP_VLD Backup Domain Valid Status Bit

Note: Hardware Cleared by Reset and Power Management Unit Whenever Backup power domain is lost. Software (SW) Set by Backup Domain Initialization Code in BOOT ROM when completed.

Value	Description
0	Backup “BKUP” Power Domain has encountered a power loss and contents are not valid.(DEFAULT)
1	Backup “BKUP” Power Domain has not encountered a power loss. Contents are valid.

Bit 30 – SRAM_VLD SRAM VALID Status Bit

Notes:

1. Hardware Clear by Reset and Power Management Units whenever SRAM power domain is lost.
2. Should be set by Users SW Code once valid data has been written into System SRAM when "MCRAMC.CTRLA.ENABLE = 0" after SRAM is initialized by user.

Value	Description
0	SRAM has encountered a power loss and contents are not valid. (DEAFULT)
1	SRAM has not encountered a power loss.

Bits 27:24 – AVREGSTDBY[3:0] Additional Voltage Regulator Configuration

Value	Name	Description
0x0	OFFINSTDBY	Regulator is OFF while in sleep mode equal or deeper than standby mode. It is OFF in hibernate and backup mode as well.
0x1	ONINSTDBY	Regulator is ON in Standby mode if AVREGEN bit is set. is OFF from Hibernate mode. It is OFF in backup mode as well.

Bits 19:16 – AVREGEN[3:0] Additional Voltage Regulator Enable

Value	Description
0x0	USB and PLL regulators disabled (Default)
0x1	USB and PLL Regulator Enabled

Bits 15:14 – ULDOLEVEL[1:0] User LDO Voltage Level Selection

Value	Name	Description
0x0	1p2v	Vout = 1.2v
0x1	1p5v	Vout = 1.5v
0x2	1p8v	Vout = 1.8v
0x3	2p5v	Vout = 2.5v

Bit 13 – ULDOSTDBY User LDO Voltage Regulator Configuration

Value	Name	Description
0x0	OFFINSTDBY	Regulator is OFF while in sleep mode equal or deeper than standby mode. It is OFF in hibernate and backup mode as well.
0x1	ONINSTDBY	Regulator is ON in Standby mode. is OFF from Hibernate mode. It is OFF in backup mode as well.

Bit 12 – ULDOEN User LDO Voltage Regulator Enable

Value	Description
0x0	User LDO is disabled
0x1	User LDO is enabled

Bits 11:8 – CPEN[3:0] Charge Pump Enable and Auto-enable.

Value	Description	Requirements
0x0	All charge pumps disabled.	AVDD ≥ 2.5v
0x1	Enable charge pump for I/O analog mux and Analog Comparator (AC)	AVDD < 2.5v
---	Reserved	
0x3	Enable charge pumps for I/O, AC, ADC	

Notes:

1. When AVDD < 2.5v the corresponding appropriate CPEN must be enabled.
2. User must have previously enabled the charge pump clocks defined in Configuration Register 5, FUCFG5.

Bit 7 – SCAPEN Super Capacitor Charging Enable

Value	Description
0	
1	

Bit 5 – LVHIB Low Voltage Hibernate Enable

Value	Name	Description
0x0	1p2v	In Hibernate mode, VDDCORE_BU and VDDCORE_RAM are set to 1.2v.

Value	Name	Description
0x1	0p8v	In Hibernate mode, VDDCORE_BU and VDDCORE_RAM are set to 0.8v

Bit 4 – LVSTDBY Low Voltage Standby Enable

Value	Name	Description
0x0	1p2v	In standby mode, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW and optionnaly VDDCOREUSB/PLL are set to 1.2v.
0x1	0p8v	In standby mode, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW and optionnaly VDDCOREUSB/PLL are set to 0.8v.

Bit 2 – OFFSTDBY Off in Standby Control for VREGSW[N-1]. Useful for Riverside only.

Value	Name	Description
0x0	OFF	In standby mode, VREGSW1,2,3 are OFF
0x1	ON	In standby mode, VREGSW1,2,3 are ON

Bits 1:0 – VREGOUT[1:0] VREG Output Control in RUN mode only. Enable by production fuse by CALSUPC.VREGOUTEN

Value	Name	Description
0x0	1p2v	In Active mode, VDDCORE_RAM, VDDCORE_BU, VDDCORE_SW and optionnaly VDDCORE_PLL USB are set to 1.2v.
0x1	1p0v	In Active mode, VDDCORE_RAM, VDDCORE_BU, VDDCORE_SW and optionnaly VDDCORE_PLL USB are set to 1.0v.
0x2	0p8v	In Active mode, VDDCORE_RAM, VDDCORE_BU, VDDCORE_SW and optionnaly VDDCORE_PLL USB are set to 0.8v.

27.7.9 VREF Control

Name: VREFCTRL
Offset: 0x020
Reset: 0x00000002
Property: RW

Table 27-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				TSEN			LPHIB	LPSTDBY
Access				RW			RW	RW
Reset				0			1	0

Bit 4 – TSEN Temperature Sensor Output Enable

Value	Description
0	Temperature sensor output to ADC disabled
1	Temperature sensor output to ADC Enabled

Bit 1 – LPHIB Bandgap and Regulators Low Power Hibernate Enable

Value	Name	Description
0x0	FullPower	In hibernate mode, bandgap is set to nominal power mode. As a consequence, enabled regulator(s) are set to nominal power mode.
0x1	LowPower	In hibernate mode, bandgap is set to low power mode. As a consequence, enabled regulator(s) are set to low power mode.

Bit 0 – LPSTDBY Bandgap and Regulators Low Power Standby Enable

Value	Name	Description
0x0	FullPower	In standby mode, bandgap and enabled regulator(s) are set to nominal power mode.
0x1	LowPower	In standby mode, bandgap and enabled regulator(s) are set to low power mode.

27.7.10 Battery Backup Power Switch. This register is implemented only if SUPC_VBAT_IMPLEMENTED==1

Name: BBPS
Offset: 0x024
Reset: 0x00000000
Property: RW

Table 27-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			SELVBATMON	ENVBATMON		WAKEEN	WKONVBATEN	CONF
Access			RW	R/W		RW	RW	RW
Reset			0	0		0	0	0

Bit 5 – SELVBATMON Select VBATMON supply voltage. 0 for VDDA, and 1 for VDDBAT.

Value	Description
0x0	VDDA supply is used for VBAT monitoring
0x1	VDDBAT is used for VBAT monitoring

Bit 4 – ENVBATMON Enables battery monitor

Value	Description
0x0	Battery Monitor is disabled
0x1	Battery Monitor is enabled

Bit 2 – WAKEEN Wake Enable

Value	Name	Description
0x0	DIS	When exiting battery backup mode, the device goes to backup mode.
0x1	EN	When exiting battery backup mode, the device is waken-up.

Bit 1 – WKONVBATEN Wake on VBAT Enable, from Backup mode

Value	Description
0x0	Remain in backup mode when VBAT is enabled
0x1	Wake from backup mode when VBAT is enabled

Bit 0 – CONF Battery Backup Power Switch Configuration

Value	Name	Description
0x0	BOR	The power switch is automatically handled by the BOR
0x1	VBAT	In Backup mode, the power-switch is forced to VBAT.

27.7.11 Backup Output Control

Name: BKOUT
Offset: 0x028
Reset: 0x00000000
Property: RW

Note: These register bits are useful for indicating to users external application logic MCU Backup mode entry/exit.

Table 27-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							TGLOM[1:0]	
Access							RW	RW
Reset							0	0
Bit	23	22	21	20	19	18	17	16
								SET
Access								W
Reset								0
Bit	15	14	13	12	11	10	9	8
								CLR
Access								W
Reset								0
Bit	7	6	5	4	3	2	1	0
								EN
Access								RW
Reset								0

Bits 25:24 – TGLOM[1:0] Toggle Output Mode

Value	Name	Description
0x0	DISABLE	The output does not toggle.
0x1	RTCTGL	The output toggles on RTC event.
0x2	BKUPTGL	The output is set when the device enters backup mode or battery backup mode. The output should then be cleared by software.

Bit 16 – SET Set Output

Bit 8 – CLR Clear Output

Bit 0 – EN Enable Output

27.7.12 Backup Input Control

Name: BKIN
Offset: 0x02C
Reset: 0x00000000
Property: R

Table 27-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					BKIN[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bits 3:0 – BKIN[3:0] Backup Input Value

28. Power Manager (PM)

28.1 Overview

The Power Manager (PM) controls the sleep modes and the power domain gating of the device

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources (SYSRST is not enabled in sleep mode) can restore the device from a sleep mode to active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The application configures which power domains are on and off during standby and hibernate sleep modes. When the application selects to enter a sleep mode, hardware automatically enters that mode when it detects no activity. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption.

In hibernate mode, the PM allows retaining the state of the I/O lines, preventing I/O lines from toggling during wake-up.

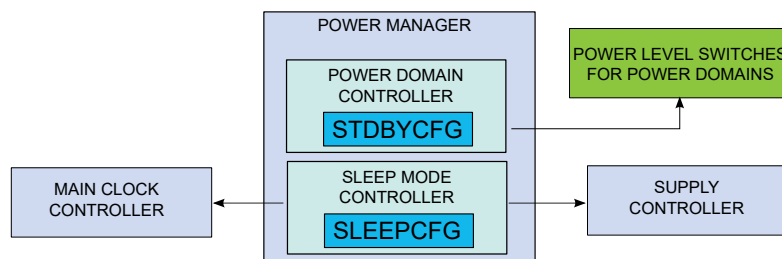
28.2 Features

The Power management module controls:

- Sleep modes: Idle, Hibernate, Standby and Off
- Two performance levels
- SleepWalking available in standby mode.

28.3 Block Diagram

Figure 28-1. Power Manager Block Diagram



28.4 Functional Description

28.4.1 Power Domains

Leaving aside the domains supplied by external power pins, such as VDDREG, AVDD, VDDIO, the device is split into these internal power domains:

- VDDCORE_BU_PD
- VDDCORE_SW_PD

- VDDCORE_RAM_PD

VDDCORE_SW_PD: is the Switchable Power Domain. It contains the CPU and all the peripherals except those located in the backup power domain.

VDDCORE_BU_PD: Contains the backup peripherals: OSC32KCTRL, SUPC, RSTC, RTC, TRAM, and the PM itself.

VDDCORE_RAM_PD: Contains the system RAM. It can be partially or fully turned off in Standby or Hibernate mode according to the user configuration.

28.4.2 Sleep Modes

The device can be set in a Sleep mode. In Sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the Sleep mode depth:

- **Active mode:** CPU is running, all power domains are active.
- **Idle sleep mode:** The CPU is stopped. Synchronous clocks are stopped except when requested. The logic state is retained.
- **Standby sleep mode:** The CPU is stopped as well as the peripherals. The logic state is retained, and power domain gating can be used to turn off the PD_CORE_RAMx power domain.
- **Hibernate sleep mode:** PD_CORE_SW power domain is turned off. The PD_CORE_RAMx power domain can be retained according to software configuration.
- **Off sleep mode:** The entire device is powered off.

28.4.3 Principle of Operation

In Active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller allows to save power by choosing between different sleep modes depending on application requirements, see *Sleep Mode Controller*.

The PM Power Domain Controller allows to reduce the power consumption in Standby mode even further.

28.4.4 Basic Operation

28.4.4.1 Initialization

After a Power-on Reset, the PM is enabled, the device is in Active mode, and all the power domains are in active state.

28.4.4.2 Enabling, Disabling, and Resetting

The PM is always enabled and cannot be reset.

28.4.4.3 Sleep Mode Controller

A Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE[2:0]) select the level of the sleep mode.

Note: Due to clock domain synchronization a small latency occurs between the store instruction and actual writing of the SLEEPCFG.SLEEPMODE[2:0]. Software must ensure that the SLEEPCFG register reads the desired value of SLEEPMODE before issuing the WFI instruction.

Entry and exit from the various sleep modes is shown in the following table:

Notes:

1. Asynchronous: interrupt generated on generic clock, external clock, or external event.
2. Synchronous: interrupt generated on synchronous (APB or AHB) clock.
3. Synchronous interrupt only for peripherals configured to run in standby.

Note: The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section

The sleep modes (Idle, Standby, Hibernate, and Off) and their effect on clock activity, regulators, and NVM state are described in the sections below. Refer to the *Power Domain Controller* section for the power domain gating effect.

Notes:

1. Stop except if running during Sleepwalking mode.
2. RiR = Run if Requested.
3. RiRR = Run if Requested or RUNSTDBY = 1.
4. LPM = Low-Power Mode.
5. On or Off, controlled by VREGCTRL.AVREGEN.
6. On or Off, controlled by (VREGCTRL.AVREGSTDBY = 1) && VREGCTRL.AVREGEN. If On then it will be in Low-Power mode.

28.4.4.3.1 Idle Mode

The IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in Active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

Entering Idle mode: The Idle mode is entered by executing the WFI instruction. Before entering the Idle mode, the user must select the Idle Sleep mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE = IDLE).

Additionally, if the SLEEPONEXIT bit in the Arm Cortex System Control register (SCR) is set, the Idle mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine. See Arm Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs.

Exiting Idle mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry according to the PRIMASK register. The system goes back to the Active mode. The CPU and affected modules are restarted.

GCLK clocks, regulators, and RAM are not affected by the Idle Sleep mode and operate in normal mode.

28.4.4.3.2 Standby Mode

The Standby mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running Sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in Sleepwalking tasks. This is the case when for example the SERCOM.CTRLA.RUNSTDBY bit is written to '1'.

Entering STANDBY mode: This mode is entered by executing the WFI instruction after writing the Sleep mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE = STANDBY).

Additionally, if the SLEEPONEXIT bit in the Arm Cortex System Control register (SCR) is set, the Standby mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine. This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Refer to the "Arm Cortex documentation" for details.

Exiting Standby mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is awakened, the device will either execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU. In Sleepwalking, a peripheral involved

in the Sleepwalking task can wake up the system whatever the interrupt type (asynchronous or synchronous).

Note: While sleep-walking, a peripheral involved in the sleepwalking task can wake up the system regardless of the interrupt type (asynchronous or synchronous).

28.4.4.3.3 Hibernate

Hibernate allow the lowest power consumption aside from off. The device is entirely powered off except for the VDDCORE_BU_PD domain. In Hibernate mode, the VDDCORE_RAM_PD power domain can be retained according to software configuration. All peripherals in backup domain are allowed to run, for example, the RTC can be clocked by a 32.768 kHz oscillator. All PM registers are reset except the CTRLA.IORET bit.

Entering Hibernate mode: This mode is entered by executing the WFI instruction after selecting the Hibernate or Backup mode by writing the Sleep Mode bits in the Sleep Configuration register (SLEEPMODE = HIBERNATE).

Exiting Hibernate or Backup mode: This mode is triggered when a Hibernate or Backup Reset is detected by the Reset Controller (RSTC).

Notes:

1. In Hibernate mode, the VREGGRAM (in Low-Power mode) regulator is used to allow powering the VDDCORE_RAM_PD power domain which can be fully retained according to software configuration.
2. When WFI instruction is executed, before entering Hibernate , the PM waits until all the synchronous clocks have been properly turned off by MCLK and MCLK honors all clock requests from peripherals before turning off the synchronous clocks. Once there are no clock requests pending, the VDDCORE_SW_PD power domain can be turned off. Consequently, software is responsible to disable peripherals before executing the sleep instruction to avoid pending clock requests.
3. Let us say an interrupt occurs at the moment when the CPU is sleeping and MCLK is honoring a pending clock request then the device will wake up and the CPU can service the corresponding interrupt handler.

28.4.4.3.4 Off Mode

In Off mode, the device is entirely powered off.

Entering Off mode: This mode is entered by selecting the Off mode in the Sleep Configuration register by writing the Sleep Mode bits (SLEEPCFG.SLEEPMODE = OFF), and subsequent execution of the WFI instruction after it is verified that the SLEEPMODE field has been set to Off mode.

Exiting Off mode: This mode is left by pulling the RESET pin low.

28.4.4.4 I/O Line State Retention in Hibernate

When entering Hibernate , PORTs are powered off but the pin configuration is retained. When the device exits the Hibernate , the I/O line configuration can either be released or stretched, based on the I/O Retention bit in the CTRLA register (CTRLA.IORET).

If IORET = 0 when exiting Hibernate , the I/O lines configuration is released and driven by the reset values of each PORT.

If the IORET = 1 when exiting Hibernate , the configuration of the I/O lines is retained until the IORET bit is written to 0. It allows the I/O lines to be retained until the application has reprogrammed each PORT.

28.4.4.5 Power Domain Controller

The Power Domain Controller provides several ways of how power domains are handled while the device is in

Standby mode:

When running a Sleepwalking task, VDDCORE_RAM_PD power domain is active whatever the STDBYCFG.RAMCFG is set to retain all RAM memory.

Dedicated power domain VDDCORE_USB_PD powered by the associated regulator can be ON or OFF according to the SUPC.VREGCTRL.AVREGSTDBY[0] and AVREGCFGEN[0] values.

Hibernate mode:

When entering a Hibernate mode, the VDDCORE_SW_PD power domain is off. As in Standby mode, the VDDCORE_RAM_PD power domain can be selectively turned ON or OFF by using HIBCFG.RAMCFG field.

OFF mode:

When entering Off mode, all the power domains are off. I/Os are in high-impedance mode except the RESET_N pad which is still in input mode able to detect a reset to wake up the chip.

28.4.4.6 Regulators, RAMs, and NVM State in Sleep Modes

Idle sleep mode:

GCLK clocks, regulators, NVM, and RAM are not affected in Idle sleep mode and operate normally.

Standby sleep mode:

By default, in Standby sleep mode, the RAMs, NVM, and regulators are automatically set in Low-Power mode in order to reduce power consumption.

- The RAM can be disabled when STDBYCFG.RAMCFG = 0 when the device is in Standby mode. RAM contents are retained.
- If not powered-down, the RAM can be set in Low-Power mode according to STDBYCFG.LPRAM.
- If neither powered-down nor Low-Power mode is enabled, the RAM data is retained and not set in retention mode. In this case, the wakeup-time is better as there is no need to turn OFF/ON the RAM power-switches.

Hibernate sleep mode:

RAM can be disabled in Hibernate mode and data retained.

If not powered-down, the RAM can be set in low-power mode according to the HIBCFG.LPRAM bit. If in low-power mode, RAM data is retained.

If neither powered-down nor low-power mode, RAM data is retained and not set in retention mode. In this case, the wakeup-time is better as there is no need to turn OFF/ON the RAM power-switches.

In Hibernate mode NVM is powered down.

28.4.5 Advanced Features

28.4.5.1 Sleepwalking

SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from Standby sleep mode. At the end of the sleepwalking task, the device can either be awakened by an interrupt (from a peripheral involved in SleepWalking) or enter again into Standby sleep mode.

In Standby when Sleepwalking is ongoing:

- All the power domains are turned ON including PDRAM power domain
- Low power mode of the regulators is not activated during sleepwalking

28.4.5.2 Wake-Up Time

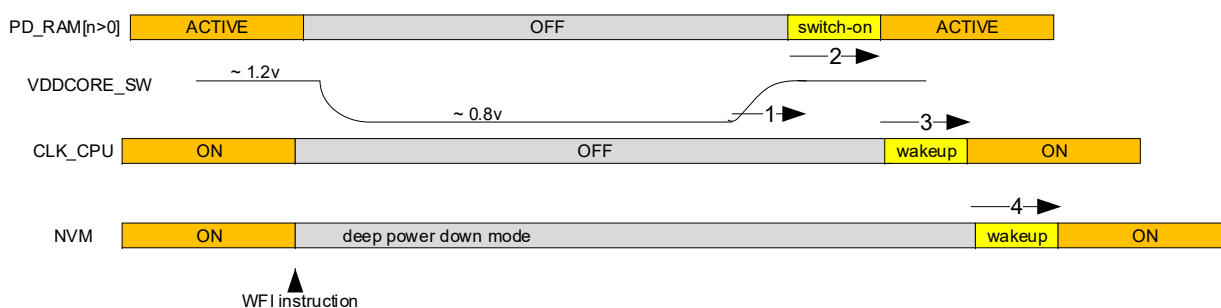
As shown in the following figure, total wake-up time depends on:

- Latency due to Reference and Regulator effect.

- Latency due to Power Domain Gating:
Usually, wake-up time is measured with the assumption that the power domains are already in active state. When using Power Domain Gating, changing a power domain from OFF to active state will take a certain time, refer to Electrical Characteristics. If all power domains were already in active state in standby sleep mode, this latency is zero.
- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM wake-up time from deep power down mode.

Figure 28-2. Total Wake-Up Time from Standby Sleep Mode

- 1: Latency due to reference wakeup (low-power mode to full-power mode) and regulator up voltage scaling.
2: Latency due to VDDCORE_RAM_PD switching ON.
3: Latency due to clock source wakeup time.
4: Latency due to NVM wakeup time from deep power down mode.



28.4.5.3 Interrupts

Sleep Mode Entry Ready (SLEEPRDY): indicates that the device is ready to enter backup sleep mode, that is, it indicates that the low power bandgap of the LPVREGC regulator is ready, given the fact that it can take several 100's of microseconds at power-up for the low power bandgap to settle.

This interrupt is a synchronous wake-up source. For example, application can setup the device to go to standby sleep and get a wakeup interrupt from SLEEPRDY

This interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. This interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled in the NVIC module. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

28.4.5.4 Sleep Mode Operation

The Power Manager is always active in sleep modes.

28.4.5.5 Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If the STANDBY sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

When there is no debug session, in the Hibernate sleep mode the core power domains are OFF. If the Hibernate sleep mode is requested by the system while in debug mode, the core domains are

kept on, and the debug modules are kept running to allow the debugger to access internal registers. When exiting the Hibernate sleep mode upon a reset condition, the core domains are reset except the debug logic, allowing users to continue in the current debug session.

Hot plugging in standby mode is supported

Hot plugging in Hibernate mode is not supported as the DSU module is not powered.

Cold plugging in Hibernate mode is supported if the external reset duration is superior to the corresponding sleep mode wakeup time. (See Electrical characteristic chapter).

28.4.5.6 Register Access Protection

Registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

28.5 Peripheral Dependencies

Table 28-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
PM	0x45000000	2 : SLEEPDLY	APB: CLKMSK[0] MASK3	---	3 STATUS0 PERID3			PD_CORE_SW

28.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0						IORET		
0x01	SLEEPCFG	7:0						SLEEPMODE[2:0]		
0x02	Reserved									
...										
0x03										
0x04	INTENCLR	7:0								SLEEPRDY
0x05	INTENSET	7:0								SLEEPRDY
0x06	INTFLAG	7:0								SLEEPRDY
0x07	Reserved									
0x08	STDBYCFG	7:0						LPRAM		RAMCFG
0x09	HIBCFG	7:0						LPRAM		RAMCFG

28.6.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: RW

Table 28-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						IORET		
Access						RW		
Reset						0		

Bit 2 – IORET I/O Retention

When entering Hibernate or Backup mode, I/O ports are powered off but the pin configurations are retained. When the device exits either mode, pin configurations are either released or stretched, based on this bit:

Value	Name	Description
0x0	NOIORET	When the device exits the HIBERNATE or BACKUP mode, the I/O line configuration are released.
0x1	IORET	When the device exits the HIBERNATE or BACKUP mode, the I/O line configuration are stretched.

28.6.2 Sleep Configuration

Name: SLEEP_CFG
Offset: 0x01
Reset: 0x02
Property: RW

Table 28-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						SLEEPMODE[2:0]		
Access						RW	RW	RW
Reset						0	1	0

Bits 2:0 – SLEEPMODE[2:0] Sleep Mode

Value	Definition
0x0	Reserved
0x1	Reserved
0x2	CPU, AHB and APB clocks are OFF (IDLE)
0x3	Reserved
0x4	All Clocks are OFF (STANDBY)
0x5	Backup domain is ON as well as some PDRAMs (HIBERNATE)
0x6	Only Backup domain is powered ON (BACKUP)
0x7	All power domains are powered OFF (OFF)

Value	Name	Description
0	IDLE0	CPU clock is OFF. This value is not public.
1	IDLE1	AHB clock is OFF. This value is not public.
2	IDLE	CPU, AHB and APB clocks are OFF
4	STANDBY	All Clocks are OFF
5	HIBERNATE	Backup domain is ON as well as some PDRAMs
6	BACKUP	Only Backup domain is powered ON
7	OFF	All power domains are powered OFF

28.6.3 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x04
Reset: 0x00
Property: RW

Table 28-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								SLEEP RDY
Access								W
Reset								0

Bit 0 – SLEEP RDY Backup Sleep Mode Entry Ready Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will disable the SLEEP RDY interrupt.

Note: Reading this bit returns whether this interrupt is enabled (1 = enabled).

28.6.4 Interrupt Enable Set

Name: INTENSET
Offset: 0x05
Reset: 0x00
Property: RW

Table 28-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								SLEEP RDY
Access								
Reset								0

Bit 0 – SLEEP RDY Backup Sleep Mode Entry Ready Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable the SLEEP RDY interrupt.

Note: Reading this bit returns whether this interrupt is enabled (1 = enabled).

28.6.5 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x06
Reset: 0x00
Property: RW

Note: Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

Table 28-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								SLEEP RDY
Access								RW
Reset								0

Bit 0 – SLEEP RDY Backup Sleep Mode Entry Ready

Note: Writing a '1' to this bit will clear the flag and the interrupt.

Value	Description
0	Device is not ready to enter Backup Sleep Mode.
1	Device is ready to enter Backup Sleep Mode.

28.6.6 Standby Configuration

Name: STDBYCFG
Offset: 0x08
Reset: 0x04
Property: RW

Table 28-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						LPRAM		RAMCFG
Access						R/W		RW
Reset						1		0

Bit 2 – LPRAM Low Power RAM Enable

Value	Description
0x0	RAM is not retained in standby when Low Power mode is active
0x1	RAM is retained in standby when Low Power mode is active

Bit 0 – RAMCFG Ram Configuration

Value	Name	Description
0	RET	All the RAMs are retained
1	OFF	Only the first 32K bytes are retained

28.6.7 Hibernate Configuration

Name: HIBCFG
Offset: 0x09
Reset: 0x04
Property: RW

Table 28-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						LPRAM		RAMCFG
Access						R/W		RW
Reset						1		0

Bit 2 – LPRAM Low Power RAM Enable

Value	Description
0x0	RAM is not retained in standby when Low Power mode is active
0x1	RAM is retained in standby when Low Power mode is active

Bit 0 – RAMCFG Ram Configuration

Value	Name	Description
0x0	RET	All the RAMs are retained
0x1	OFF	Only the first 32K bytes are retained

29. Reset Controller (RSTC)

29.1 Overview

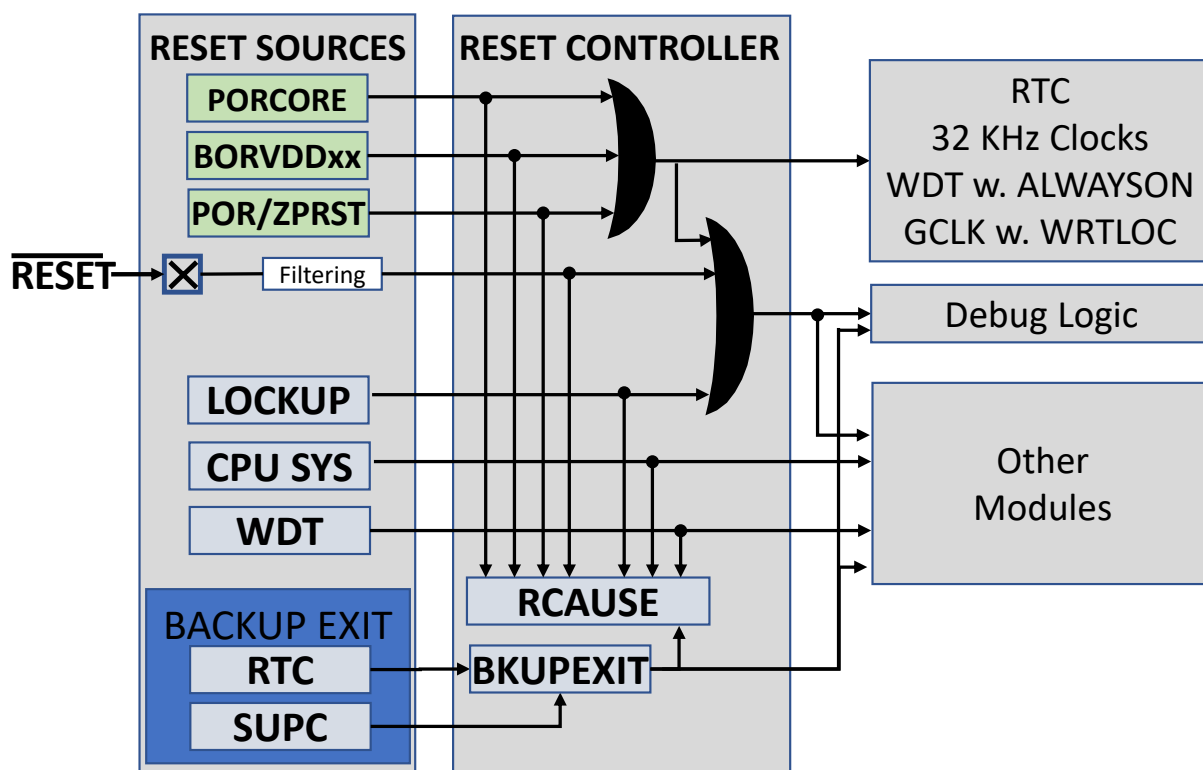
The Reset Controller (RSTC) manages the reset of the Microcontroller. It issues a Microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

29.2 Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources
 - **Power supply reset sources:** POR, PORCORE, BOR33
 - **User reset sources:** External reset (RESET), Watchdog reset, System Reset Request, CPU Lockup reset
 - **Backup exit sources:** Real-Time Counter (RTC)

29.3 Block Diagram

Figure 29-1. Reset Controller Block Diagram



29.4 Signals Description

Table 29-1. Signals Description

Signal Name	Type	Description
RESET	Digital input	External reset

29.5 Functional Description

The Reset Controller collects the various Reset sources and generates Resets for the device.

29.5.1 Basic Operation

29.5.1.1 Initialization

After a Power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the Reset source.

29.5.1.2 Enabling, Disabling, and Resetting

The RSTC module is always enabled.

29.5.1.3 Reset Causes and Effects

The latest Reset cause is available in the RCAUSE register and can be read during the application boot sequence to determine proper action.

These are the groups of Reset sources:

- **Power supply Reset:** Resets caused by an electrical issue. It covers POR Resets
- **Brown-out Reset:** Resets caused by BORVDDx detectors, PORCORE and BOD33
- **User Reset:** Resets caused by the application. It covers external Resets, system Reset requests, CPU Lockup reset and watchdog Resets.
- **Backup Reset:** Reset cause when leaving backup or hibernate mode due to wake event RTC, BBPS, VBAT, or $\overline{\text{RESET}}$

The following table lists the parts of the device that are reset, depending on the Reset type.

Table 29-2. Effects of Reset Causes

Effect: Reset:	CPU Logic, Bus Logic	RTC ⁽¹⁾ , OSC32KCTRL, BKOUT, BOR ⁽²⁾ , SUPC, VREGCTRL & VREFCTRL ⁽³⁾ registers, CTRLA bits of PM	Flash Panel Controller	Debug Logic	Other Modules
POR, PORCORE	Reset	Reset	Reset	Reset	Reset
External Reset BORVDDx	Reset	No Reset	No Reset	Reset	Reset
WDT Reset Lockup Reset System Reset Request	Reset	No Reset	No Reset	No Reset	Reset
RTC, BBPS, VBAT	-	No Reset	Reset	Reset	-

Notes:

1. RTC.DBGCTRL is reset in Backup Sleep mode.
2. The BOR register is retained during Backup mode and is not reset when exiting.
3. VREGCTRL.VREGOUT is reset by User Reset so regulators go back to default setting.

The external Reset is generated when pulling the RESET pin low.

The POR, PORCORE, and BORVDDx (for example, BORVDDA, BORVDDREG) Reset sources are generated by the analog modules inside the PWR_SMOR or PWR_SMOR_DS controlled by the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for additional information, refer to the “Arm® Cortex™ Technical Reference Manual” which is available for download at www.arm.com).

The Lockup Reset is a Reset generated by the CPU when it enters a lockup state (for additional information, refer to the “Arm® Cortex™ Technical Reference Manual” which is available for download at www.arm.com).

29.5.2 Power Management

The Reset Controller module is always on.

29.5.3 Sleep Mode Operation

The RSTC module is active in all sleep modes.

29.5.4 Debug Operation

When the CPU is halted in Debug mode, the RSTC continues normal operation.

If the DBGCTRL.LCKUPDIS bit is set, the CPU LOCKUP reset source is disabled in Debug mode. Refer to the DBGCTRL register for additional information.

29.5.5 PAC Write-Protection

All registers with write-access can be optionally write protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description.

When the CPU is halted in Debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

29.6 Peripheral Dependencies

Table 29-3. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
RSTC	0x45004000		APB: CLKMSK[0] MASK5	---	5 STATUS0 PERID5			PD_CORE_SW

29.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	RCAUSE	15:8							LOCKUP	BACKUP
		7:0	SYST	WDT	EXT	BORVDDIO	BORVDDA	BORVDDREG	PORCORE	POR
0x02	BKUPEXIT	7:0	HIB						RTC	
0x03	Reserved									
0x04	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								LCKUPDIS

29.7.1 Reset Cause

Name: RCAUSE
Offset: 0x0
Reset: None
Property: R

Note: Only one bit is set at any one time. This register only reflects the last event.

Table 29-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
							LOCKUP	BACKUP
Access							R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	SYST	WDT	EXT	BORVDDIO	BORVDDA	BORVDDREG	PORCORE	POR
Access	R	R	R	R	R	R	R	R
Reset								

Bit 9 – LOCKUP The CPU was reset because a LOCKUP Reset from main cpu has occurred.
CPU has entered lockup state. (for details, refer to the “Arm® Cortex™ Technical Reference Manual” available at www.arm.com).

Bit 8 – BACKUP The CPU was reset because a backup exit condition has occurred.
Reset cause when leaving Backup or Hibernate mode due to wake event (RTC).

Bit 7 – SYST The CPU was reset because a software reset (system reset) from main CPU has been performed
Reset generated by a System Reset Request by the CPU by asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details, refer to the “Arm® Cortex™ Technical Reference Manual” available at www.arm.com).

Bit 6 – WDT The CPU was reset due to a watchdog reset
Reset generated by Watchdog Timer.

Bit 5 – EXT The CPU was reset due to the RESET pin being asserted.
Reset by the $\overline{\text{RESET}}$ pin driven low.

Bit 4 – BORVDDIO The CPU was reset due to the VDDIO supply voltage being lower than the brown-out threshold level 3.3 V.
Reset generated by VDDIO BOR Detector.

Bit 3 – BORVDDA The CPU was reset due to the VDDA supply voltage being lower than the brown-out threshold level 3.3 V.
Reset generated by VDDA BOR Detector.

Bit 2 – BORVDDREG The CPU was reset due to the VDDREG supply voltage being lower than the brown-out threshold level 3.3 V.
Reset by VDDREG BOR Detector.

Bit 1 – PORCORE The CPU was reset due to the supply voltage being lower than the brown-out threshold level 1.2 V.

Reset by Core Power-on Reset.

Bit 0 – POR The CPU was reset due to the supply voltage being lower than the power-on level

Reset generated by Power-on Reset.

29.7.2 Backup Exit Source. Implemented only if RSTC_BACKUP_IMPLEMENTED=1

Name: BKUPEXIT
Offset: 0x2
Reset: 0x00
Property: R

Note: Backup exit only: RTC=1
Hibernate exit: HIB and RTC = 1

Table 29-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	HIB						RTC	
Access	R						R	
Reset	0						0	

Bit 7 – HIB Exit from Hibernate sleep mode
Wake up from Hibernate Sleep Mode.

Bit 1 – RTC Backup reset condition is RTC interrupt
Wake up from Real Time Counter (RTC) interrupt.

29.7.3 Debug Control

Name: DBGCTRL
Offset: 0x4
Reset: 0x00000000
Property: RW

Table 29-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								LCKUPDIS
Access								RW
Reset								0

Bit 0 – LCKUPDIS Lockup Disable

If set the CPU LOCKUP Reset source is disabled in debug mode.

(The Lockup Reset is a Reset generated by the CPU when it enters a lockup state (for details refer to the ARM® Cortex™ Technical Reference Manual on www.arm.com)).

30. External Interrupt Controller (EIC)

30.1 Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, on falling, on both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event.

A separate non-maskable interrupt (NMI) is also supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other interrupt mode.

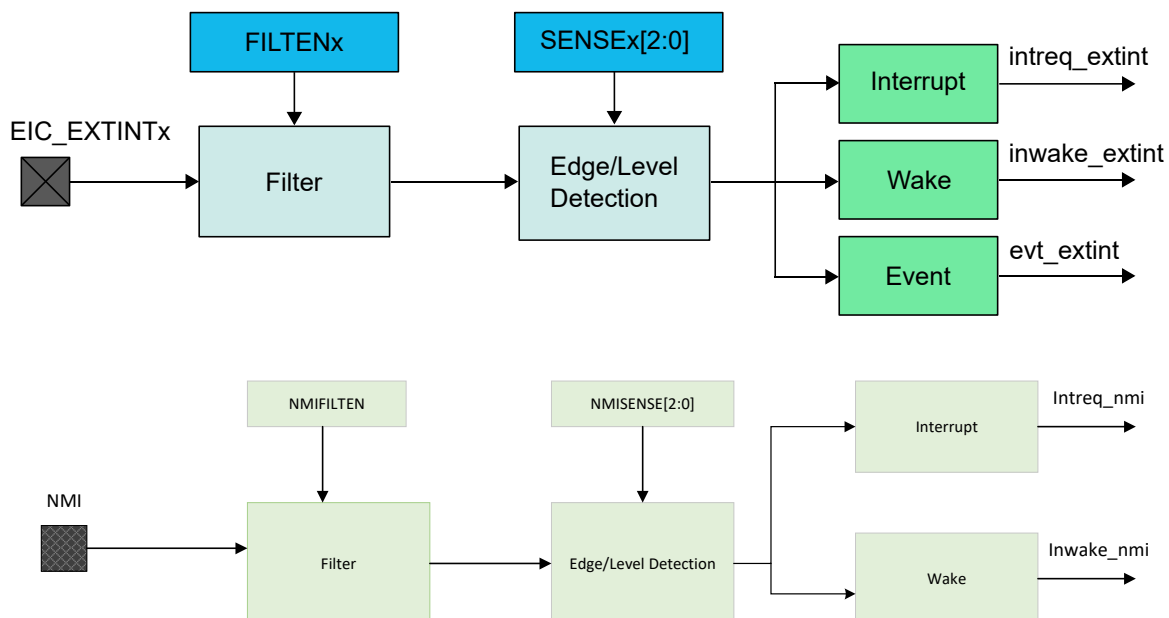
30.2 Features

The following are key features of the module:

- Up to 16 external interrupt pins (EIC_EXTINTx, x = 0-15)
- Dedicated, individually maskable interrupt for each pin
- Interrupt on rising, falling, or both edges
- Synchronous or asynchronous edge detection mode
- Interrupt pin debouncing
- Interrupt on high or low levels
- Asynchronous interrupts for sleep modes without clock
- Filtering of external interrupt, EIC_EXTINTx, pins
- Event generation trigger events from EIC_EXTINTx pins

30.3 Block Diagram

Figure 30-1. EIC Block Diagram



30.4 Signal Description

Signal Name	Type	Description
EIC_EXTINTx[15..0]	Digital Input	External interrupt pin
NMI	Digital Input	Non-maskable interrupt pin

Note: One signal may be available on several pins.

30.5 Interrupts

The EIC has the following interrupt sources: External interrupt pins (EIC_EXTINTx; x = 0-15).

Each interrupt source has an associated interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an interrupt condition occurs. Each interrupt, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET = 1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR = 1).

An interrupt request is generated when the interrupt flag is set, and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. Refer to the INTFLAG register for additional information on how to clear interrupt flags. The EIC has at least one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

30.6 Events

The EIC can generate the following output events:

- External event from pin EIC_EXTINTx; x= 0-15

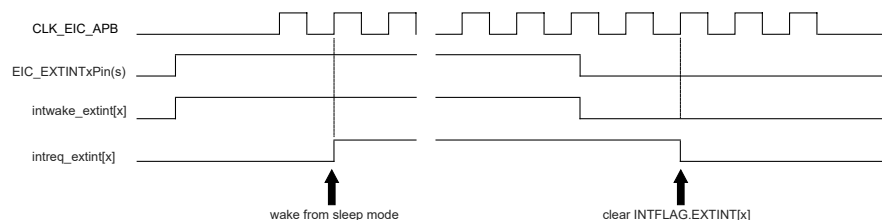
Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EIC_EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

30.7 Sleep Mode Operation

In sleep modes, an EIC_EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the CONFIG register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.

Figure 30-2. Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)



30.8 Secure Access Right

Non-secure write to CTRLA register or DPRESALER register is prohibited. Non-secure read to CTRLA or DPRESALER register or SYNCBUSY register will return zero with no error resulting. Non-secure write to a bit of EVCTRL, ASYNCH, DEBOUNCEN, INTENCLR, INTENSET, INTFLAG and CONFIG registers is prohibited if the related bit of NONSEC.EXTINT is zero. Non-secure write to NMICTRL and NMIFLAG registers is prohibited if NONSECNMI.NMI is zero. Bits relating to secure EXTINT read as zero in non-secure mode with no error resulting.

30.9 Synchronization

Due to the main clock domain and the peripheral clock domains being asynchronous, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register(CTRLA.SWRST)
- Enable bit in control register(CTRLA.ENABLE)

Required write-synchronization is denoted by the "Write-Synchronized" property in the register description.

30.10 Peripheral Dependencies

Table 30-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
EIC	0x44020000	NMI : NMI 16 : EXTINT0 17 : EXTINT1 18 : EXTINT2 19 : EXTINT3 20 : EXTINT4 21 : EXTINT5 22 : EXTINT6 23 : EXTINT7 24 : EXTINT8 25 : EXTINT9 26 : EXTINT10 27 : EXTINT11 28 : EXTINT12 29 : EXTINT13 30 : EXTINT14 31 : EXTINT15 32 : NSCHK	APB: CLKMSK[0] MASK13	PCHCTRL[5]	13 STATUS0 PERID13		Generators: EXTINT0: CHANNEL[21] EXTINT1: CHANNEL[22] EXTINT2: CHANNEL[23] EXTINT3: CHANNEL[24] EXTINT4: CHANNEL[25] EXTINT5: CHANNEL[26] EXTINT6: CHANNEL[27] EXTINT7: CHANNEL[28] EXTINT8: CHANNEL[29] EXTINT9: CHANNEL[30] EXTINT10: CHANNEL[31] EXTINT11: CHANNEL[32] EXTINT12: CHANNEL[33] EXTINT13: CHANNEL[34] EXTINT14: CHANNEL[35] EXTINT15: CHANNEL[36]	PD_CORE_SW

30.11 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0				CKSEL			ENABLE	SWRST
0x01	NMICTRL	7:0				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
0x02	NMIFLAG	15:8								
		7:0								NMI
0x04	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x08	EVCTRL	31:24	EXTINTEO[31:24]							
		23:16	EXTINTEO[23:16]							
		15:8	EXTINTEO[15:8]							
		7:0	EXTINTEO[7:0]							
0x0C	INTENCLR	31:24	EXTINT[31:24]							
		23:16	EXTINT[23:16]							
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x10	INTENSET	31:24	EXTINT[31:24]							
		23:16	EXTINT[23:16]							
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x14	INTFLAG	31:24	EXTINT[31:24]							
		23:16	EXTINT[23:16]							
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x18	ASYNCH	31:24	ASYNCH[31:24]							
		23:16	ASYNCH[23:16]							
		15:8	ASYNCH[15:8]							
		7:0	ASYNCH[7:0]							
0x1C	CONFIG	31:24	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
		23:16	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
		15:8	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
		7:0	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
0x20 ... 0x2F	Reserved									
0x30	DEBOUNCEN	31:24	DEBOUNCEN[31:24]							
		23:16	DEBOUNCEN[23:16]							
		15:8	DEBOUNCEN[15:8]							
		7:0	DEBOUNCEN[7:0]							
0x34	DPRESCALER	31:24								
		23:16								TICKON
		15:8								
		7:0					STATES	PRESCALER[2:0]		
0x38	PINSTATE	31:24	PINSTATE[31:24]							
		23:16	PINSTATE[23:16]							
		15:8	PINSTATE[15:8]							
		7:0	PINSTATE[7:0]							
0x3C	NSCHK	31:24	NMI	EXTINT[30:24]						
		23:16	EXTINT[23:16]							
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							
0x40	NONSEC	31:24	NMI	EXTINT[30:24]						
		23:16	EXTINT[23:16]							
		15:8	EXTINT[15:8]							
		7:0	EXTINT[7:0]							

30.11.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00
Property: RW

Note:

1. Reserved bits must always be written as '0'.

Table 30-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
				CKSEL			ENABLE	SWRST
Access				RW			RW	W
Reset				0			0	0

Bit 4 – CKSEL Clock Selection

The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32 kHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).

This bit is not Write-Synchronized.

This bit is Enable-Protected.

Value	Name	Description
0	CLK_GCLK	Clocked by GCLK
1	CLK_ULP32K	Clocked by ULP32K

Bit 1 – ENABLE Enable

Due to synchronization there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register will be set (SYNCBUSY.ENABLE = 1). SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not Enable-Protected.

This bit is Write-Synchronized.

Note: This bit is write protected and can only be written when CTRLA.ENABLE = 0.



Important: Enable-protected bits in the CTRLA register can be written at the same time when setting CTRLA.ENABLE to '1', but not at the same time when CTRLA.ENABLE is being cleared.

Value	Description
0	The EIC is disabled.
1	The EIC is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

Value	Description
0	There is no ongoing reset operation.
1	The reset operation is ongoing.

30.11.2 Non-Maskable Interrupt Control

Name: NMICTRL
Offset: 0x01
Reset: 0x00
Property: RW



Important: On PIC32CM SG devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the NMI interrupt is set as Non-Secure in the NONSEC register (NONSEC.NMI bit).

Table 30-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
				NMIASYNCH	NMIFILTEN	NMISENSE[2:0]		
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bit 4 – NMIASYNCH Asynchronous Edge Detection Mode

The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

Value	Name	Description
0	SYNC	Edge detection is clock synchronously operated
1	ASYNCH	Edge detection is clock asynchronously operated

Bit 3 – NMIFILTEN Non-Maskable Interrupt Filter Enable

Value	Description
0	NMI filter is disabled.
1	NMI filter is enabled.

Bits 2:0 – NMISENSE[2:0] Non-Maskable Interrupt Sense Configuration

These bits define on which edge or level the NMI triggers.

Value	Name	Description
0	NONE	No detection
1	RISE	Rising-edge detection
2	FALL	Falling-edge detection
3	BOTH	Both-edges detection
4	HIGH	High-level detection
5	LOW	Low-level detection

30.11.3 Non-Maskable Interrupt Flag Status and Clear

Name: NMIFLAG
Offset: 0x02
Reset: 0x0000
Property: RW



Important: On PIC32CM SG devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the NMI interrupt is set as Non-Secure in the NONSEC register (NONSEC.NMI bit).

Table 30-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								NMI
Access								RW
Reset								0

Bit 0 – NMI Non-Maskable Interrupt

This flag is cleared by writing a '1' to it.

This flag is set when the NMI pin matches the NMI sense configuration and will generate an interrupt request.

Writing a '0' to this bit has no effect.

30.11.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x04
Reset: 0x00000000
Property: R

Table 30-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.ENABLE bit is complete.
1	Write synchronization for CTRLA.ENABLE bit is ongoing.

Bit 0 – SWRST Software Reset Synchronization Busy Status

Value	Description
0	Write synchronization for CTRLA.SWRST bit is complete.
1	Write synchronization for CTRLA.SWRST bit is ongoing.

30.11.5 Event Control

Name: EVCTRL
Offset: 0x08
Reset: 0x00000000
Property: RW

 **Important:** On PIC32CM SG devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Table 30-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EXTINTEO[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EXTINTEO[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EXTINTEO[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EXTINTEO[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – EXTINTEO[31:0] External Interrupt Event Output Enable

The bit x of EXTINTEO enables the event associated with the EIC_EXTINTx pin.

Value	Description
0	Event from pin EIC_EXTINTx is disabled.
1	Event from pin EIC_EXTINTx is enabled and will be generated when EIC_EXTINTx pin matches the external interrupt sensing configuration.

30.11.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0C
Reset: 0x00000000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

 **Important:** On PIC32CM5G devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

Table 30-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EXTINT[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EXTINT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the NSCHK Interrupt Enable bit.

Note: For PIC32CM5G devices, this bit is reserved.

Value	Description
0	The NSCHK interrupt is disabled
1	The NSCHK interrupt is enabled

Bits 31:0 – EXTINT[31:0] External Interrupt Enable

The bit x of EXTINT enables the interrupt associated with the EIC_EXTINTx pin.

Writing a '0' to bit x has no effect.

Writing a '1' to bit x will clear the External Interrupt Enable bit x, which disables the external interrupt EIC_EXTINTx pin(s).

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

30.11.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

 **Important:** On PIC32CM5G devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

Table 30-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EXTINT[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EXTINT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the NSCHK Interrupt Enable bit.

Note: For PIC32CMGC devices, this bit is reserved.

Value	Description
0	The NSCHK interrupt is disabled
1	The NSCHK interrupt is enabled

Bits 31:0 – EXTINT[31:0] External Interrupt Enable

The bit x of EXTINT enables the interrupt associated with the EIC_EXTINTx pin.

Writing a '0' to bit x has no effect.

Writing a '1' to bit x will set the External Interrupt Enable bit x, which enables the external interrupt EIC_EXTINTx pin(s).

Value	Description
0	The external interrupt EIC_EXTINTx is disabled.
1	The external interrupt EIC_EXTINTx is enabled.

30.11.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: RW



Important: On PIC32CM SG devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit).

Note:

1. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 30-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EXTINT[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EXTINT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NSCHK Non-secure Check Interrupt

This flag is set when write to either NONSEC and NSCHK register and if the related bit of NSCHK is enabled and the related bit of NONSEC is zero.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the flag.

Note: For PIC32CMGC devices, this bit is reserved.

Value	Description
0	The NSCHK interrupt flag is not set.
1	The NSCHK interrupt flag is set.

Bits 31:0 – EXTINT[31:0] External Interrupt

The flag bit x is cleared by writing a '1' to it.

This flag is set when the EIC_EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR.EXTINT[x] or INTENSET.EXTINT[x] is '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the External Interrupt x flag.

30.11.9 External Interrupt Asynchronous Mode

Name: ASYNCH
Offset: 0x18
Reset: 0x00000000
Property: RW



Important: On PIC32CM SG devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Table 30-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ASYNCH[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ASYNCH[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ASYNCH[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ASYNCH[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ASYNCH[31:0] Asynchronous Edge Detection Mode

The bit x of ASYNCH set the Asynchronous Edge Detection mode for the interrupt associated with the EIC_EXTINTx pins.

Value	Name	Description
0	SYNC	EIC_EXTINTx edge detection is synchronously operated
1	ASYNCH	EIC_EXTINTx edge detection is asynchronously operated

30.11.10 External Interrupt Sense Configuration

Name: CONFIG
Offset: 0x1C
Reset: 0x00000000
Property: RW

Table 30-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FILTEN7	SENSE7[2:0]			FILTEN6	SENSE6[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FILTEN5	SENSE5[2:0]			FILTEN4	SENSE4[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FILTEN3	SENSE3[2:0]			FILTEN2	SENSE2[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FILTEN1	SENSE1[2:0]			FILTEN0	SENSE0[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – FILTEN7 Filter Enable 7

Bits 30:28 – SENSE7[2:0] Input Sense Configuration 7

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

Bit 27 – FILTEN6 Filter Enable 6

Bits 26:24 – SENSE6[2:0] Input Sense Configuration 6

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection

Value	Name	Description
4	HIGH	High level detection
5	LOW	Low level detection

Bit 23 – FILTEN5 Filter Enable 5

Bits 22:20 – SENSE5[2:0] Input Sense Configuration 5

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

Bit 19 – FILTEN4 Filter Enable 4

Bits 18:16 – SENSE4[2:0] Input Sense Configuration 4

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

Bit 15 – FILTEN3 Filter Enable 3

Bits 14:12 – SENSE3[2:0] Input Sense Configuration 3

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

Bit 11 – FILTEN2 Filter Enable 2

Bits 10:8 – SENSE2[2:0] Input Sense Configuration 2

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

Bit 7 – FILTEN1 Filter Enable 1

Bits 6:4 – SENSE1[2:0] Input Sense Configuration 1

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

Bit 3 – FILTEN0 Filter Enable 0

Bits 2:0 – SENSE0[2:0] Input Sense Configuration 0

Value	Name	Description
0	NONE	No detection
1	RISE	Rising edge detection
2	FALL	Falling edge detection
3	BOTH	Both edges detection
4	HIGH	High level detection
5	LOW	Low level detection

30.11.11 Debouncer Enable

Name: DEBOUNCEN
Offset: 0x30
Reset: 0x00000000
Property: RW



Important: On PIC32CM SG devices where the EIC is configured as Mix-Secure, read and write accesses are allowed only if the external interrupt x (EXTINTx) is set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Some restrictions apply for the Non-Secure accesses to an Enabled-Protected register as it will not be possible for the Non-Secure to configure it once this register is enabled by the Secure application. This will require some veneers to be implemented on Secure side.

Table 30-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DEBOUNCEN[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DEBOUNCEN[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DEBOUNCEN[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DEBOUNCEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DEBOUNCEN[31:0] Debouncer Enable

The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EIC_EXTINTx pin(s).

Value	Description
0	The EIC_EXTINTx edge input is not debounced.
1	The EIC_EXTINTx edge input is debounced.

30.11.12 Debouncer Prescaler

Name: DPRESALER
Offset: 0x34
Reset: 0x00000000
Property: RW

Table 30-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								TICKON
Access								RW
Reset								0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					STATES	PRESCALER[2:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit 16 – TICKON Pin Sampler frequency selection

This bit selects the clock used for the sampling of bounce during transition detection.

Value	Name	Description
0	CLK_GCLK_EIC	Bounce sampler is using GCLK_EIC
1	CLK_LFREQ	Bounce sampler is using Low Frequency Clock

Bit 3 – STATES Debouncer number of states

Value	Name	Description
0	LFREQ3	3 low frequency samples
1	LFREQ7	7 low frequency samples

Bits 2:0 – PRESCALER[2:0] Debouncer Prescaler

Value	Name	Description
0	DIV2	EIC clock divided by 2
1	DIV4	EIC clock divided by 4
2	DIV8	EIC clock divided by 8
3	DIV16	EIC clock divided by 16
4	DIV32	EIC clock divided by 32
5	DIV64	EIC clock divided by 64

Value	Name	Description
6	DIV128	EIC clock divided by 128
7	DIV256	EIC clock divided by 256

30.11.13 Pin State

Name: PINSTATE
Offset: 0x38
Reset: 0x00000000
Property: R

 **Important:** On PIC32CM SG devices where the EIC is configured as Mix-Secure, non-secure read accesses will return the true value only for external interrupt x (EXTINTx) set as Non-Secure in the NONSEC register (NONSEC.EXTINTx bit). Else the read will return 0 for the external interrupt x (EXTINTx) set as Secure in the NONSEC register (NONSEC.EXTINTx bit).

Table 30-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PINSTATE[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PINSTATE[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PINSTATE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PINSTATE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PINSTATE[31:0] Pin State

These bits return the valid pin state of the debounced external interrupt pin, EIC_EXTINTx.

30.11.14 Non-secure Interrupt Check Enable

Name: NSCHK
Offset: 0x3C
Reset: 0x00000000
Property: RW

 **Important:** This register is only relevant for PIC32CM SG00/GC00 devices when the EIC has been defined as Mix-Secure in the H2PB bridge.

This register allows the user to select one or more external pins to check their security attribution as non-secured.

Table 30-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NMI	EXTINT[30:24]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EXTINT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NMI Non-Maskable External Interrupt Nonsecure Check Enable

This bit selects the Non-Maskable Interrupt pin for security attribution check. If the NMI bit in NONSECNMI is set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

Bits 30:0 – EXTINT[30:0] External Interrupt Nonsecure Check Enable

These bits select the individual pins for security attribution check. If any pin selected in NSCHK has the corresponding bit in NONSEC set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.

Value	Description
1	1-to-0 transition will be detected on corresponding NONSEC bit.

30.11.15 Non-secure Interrupt

Name: NONSEC
Offset: 0x40
Reset: 0x00000000
Property: RW

 **Important:** This register is only relevant for PIC32CM SG00/GC00 devices when the EIC has been defined as Mix-Secure in the H2PB bridge.

This register allows to set the NMI or external interrupt control and status registers in non-secure mode, individually per interrupt pin.

Table 30-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NMI	EXTINT[30:24]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EXTINT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EXTINT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EXTINT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – NMI Non-Maskable Interrupt Nonsecure Enable

This bit enables the non-secure mode of NMI.

The registers whose content is set in non-secure mode by NONSEC.NMI are NMICTRL and NMIFLAG registers.

Value	Description
0	NMI is secure.
1	NMI is non-secure.

Bits 30:0 – EXTINT[30:0] External Interrupt Nonsecure Enable

The bit x of EXTINT enables the non-secure mode of EXTINTx.

The registers whose EXTINT bit or bitfield x is set in non-secure mode by NONSEC.EXTINTx are EVCTRL, ASYNCH, IDEBOUNCEN, INTENCLR, INTENSET, INTFLAG and CONFIG registers.

Value	Description
0	EXTINTx is secure.
1	EXTINTx is non-secure.

31. Non-Volatile Memory Controller (NVMCTRL)

The NVM Controller is built from two separate modules, described below, the Flash Controller Write (FCW) and Flash Controller Read (FCR).

31.1 Flash Controller, Write

31.1.1 Features

The following are key feature of the Flash Controller, Write Module:

- Support Flash partitions
 - Program Flash Memory (PFM)
 - Boot Flash Memory (BFM)
 - Configuration Flash Memory (CFM)
- Mission Mode Write/Erase Features:
 - PFM Erase
 - Page Erase
 - Single Write (32-bit) + simple parity
 - Quad Write (128-bit) + SECDED ECC
 - Row Write + SECDED ECC
 - Built-in DMA of data from DRAM
 - All write operations support Pre-Program to increase endurance and retention
- BFM Address Swap
 - Supports Dual Boot
 - Support Live Update
- PFM Address Swap
- Debug Erase features are implemented by ROM Code:
 - Identical to Mission Mode – PLUS:
 - Chip Erase: PFM [+ SRF] & unprotected NVR pages - available to the Boot only
 - CFM pages must be unprotected to erase
 - NVR Test pages cannot be erased
 - BRCTRL SFR for feature lockout
- Debug Programming Features
 - Identical to Mission Mode
- Program Flash Memory (PFM) Write Protect, with 4 protection regions
 - Region defined by Base and Size
 - Lockable
- Boot Flash Memory (BFM) Write Protect
 - Individual (4KB) Page write protect regions
 - Compatible with Dual Boot
 - Lockable
- Flash based User OTP pages
 - Not erased by Chip Erase or Page Erase
 - 4 Lockable Write Protect regions per Page

- Supports storage of user calibration values
- Security features for a TrustZone (TZ) CPU
 - Intended to be owned by trusted code
 - Not Split-Secure
 - Write of Tamper Log on configured tamper detect

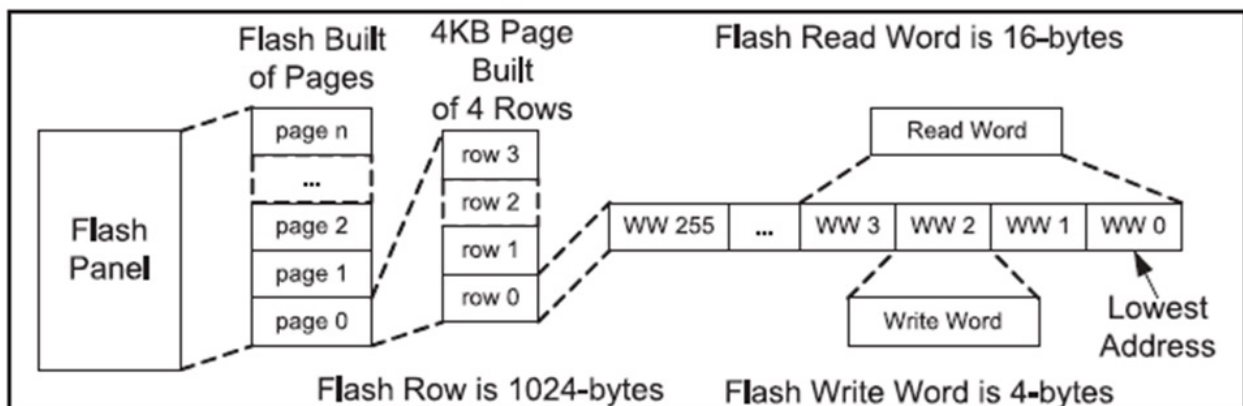
31.1.2 Definitions

The Flash Controller, Write (FCW) supports programming operations (writes/erases) for on-chip flash memory. This device's memories has the following addressing:

Table 31-1. Memory Addressing

Addressing Parameter	Value	Units
General		
Bytes per Row	1024	Bytes
Word Size - bits (Data, Reg, Fuse)	32	bits
Word Size - Bytes (Data, Reg, Fuse)	4	Bytes
Address Increment	32	bits
Standard		
Single Write + simple parity	32	bits
Double Write + simple parity	64	bits
Flash		
Single Quad Read Word (QRW)	128	bits
QRW w/ECC (12-bits)	140	bits
Single Quad Write Word (QWW)	128	bits
QWW w/ECC (3-bits)	131	bits
Quad Double Read Word (QDRW)	256	bits
QDRW w/ECC (16-bits)	272	bits
Quad Double Write Word (QDWW)	256	bits
QDWW w/ECC (4-bits)	260	bits

Figure 31-1. Flash Module Construction



Flash regions of concern include:

- Program Flash Memory (PFM)
- Boot Flash Memory (BFM)
- Configuration Flash Memory (CFM)

A Flash panel is divided into two partitions: Main and NVR. Main contains the PFM while NVR both the BFM and CFM.

Be aware that a panel is not a physical construct, but a conceptual model. (The System Address Map shows the actual memory layout.)

31.1.3 Flash Architecture

Each Flash partition is built up of several pages, also called sectors. The controller works with a panel made from 4KB pages with each page containing 4 rows of Flash data. A row is the largest selectable region for contiguous programming of Write Words. A Row contains 4 Write Words.

A page of Flash is the smallest unit of memory that can be erased in a single operation. A panel's Program Flash Memory (PFM) space can also be erased in a single operation. All other erases use multiple operations.

A Write Word is the only unit of memory that can be programmed at a time. All other programming operations are made up of several contiguous Write Word Program operations. The FCW supports:

- **Row Write:** Write Word by Write Word programming until the whole Row is written. Data is read by the FCW from System SRAM.
- **Quad Write:** The FCW performs 4 writes of data from holding registers
- **Single Write:** The FCW performs one write of data from holding register(s)

See the table in the Definitions section.

31.1.3.1 Program Flash Memory (PFM)

PFM is the largest section of Flash memory. It can span across two panels. When two Panels are used, the location for Panel 1 is in the lower address range and Panel 2 in the upper address range. The order in which each PFM exist in the Flash memory space can be controlled by software. The PFM address range is contiguous across both panels.

Dual panel systems support Live Update which allows reading from one panel while write to the other. It does not matter which of the logical regions are being accessed as long as they are in different panels.

31.1.3.2 Boot Flash Memory (BFM)

Boot Flash Memory is meant to support sophisticated boot loaders. It has 4 pages that are either divided between dual panels, or all pages are in a single BFM panel.

PIC32C devices have a Boot ROM to control configuration loading and to provide a root of trust for secure boot. When the Boot ROM has finished loading the configuration it sets up the CPU to start execution from the base address of the BFM.

The BFM can be used in either Dual Boot or Single Boot. With a dual panel system, each panel has BFM code. Dual Boot designates each panel's BFM as a boot source. This allows for the safe updating of one boot image while the other image stays intact. A Single Boot, with two panels, allows the boot image to span both panel's BFM space providing for a larger boot code space. The BFM only has the memory in the existing panel when a single panel exists.

31.1.3.3 Configuration Flash Memory (CFM)

CFM contains several pages. Like BFM it is either divided between dual panels, or all pages are in a single CFM panel. This memory is dedicated to the configuration of the device (See the following table). These pages have hardware-imposed restrictions on their usage. They are not for code and may not be accessible after the device is protected.

One page contains factory calibration data including the unique ID of the device . This page cannot be written or erased.

The Boot Configuration (Boot CFG) page contains pre-boot options to be configured by the Boot ROM. This page can be read protected and write/erase protected by the Flash system.

The User-OTP page implements a Flash-based “One Time Programmable” region. It is always erase protected by hardware such that its content survives a Chip Erase operation. It is intended to store calibration values for external devices but is fully user defined. That is it can be written only once, but accessed thereafter.

The User Configuration (User CFG) page stores pre-boot options that have a different protection model than Boot CFG. This page can be write/erase locked.

For systems that have dual panels, the second panel's CFM contains additional pages to support dual boot. Boot CFG 2 and User CFG 2 allows for safe updates of configuration data while also updating application code in PFM, or boot code in BFM. This is different than a bootloader.

Table 31-2. Flash CFM Configuration Address Map

Panel:	Address:		Size:	Contents:	Notes:
	Start:	End:			
1	0x0A00_0000	0x0A00_0FFF	4 KByte	User CFG-1	-
1	0x0A00_1000	0x0A00_1FFF	4 KByte	User OTP-1	-
1	0x0A00_2000	0x0A00_2FFF	4 KByte	BOOT CFG-1	-
1	0x0A00_3000	0x0A00_3FFF	4 KByte	ROM CFG-1	-
1	0x0A00_4000	0x0A00_4FFF	4 KByte	VSS0	-
1	0x0A00_5000	0x0A00_5FFF	4 KByte	VSS1	-
1	0x0A00_6000	0x0A00_6FFF	4 KByte	RSVD	-
1	0x0A00_7000	0x0A00_7FFF	4 KByte	CAL-OTP	Cannot be written or erased.
2	0x0A00_8000	0x0A00_8FFF	4 KByte	User CFG-2	-
2	0x0A00_9000	0x0A00_9FFF	4 KByte	User OTP-2	-
2	0x0A00_A000	0x0A00_AFFF	4 KByte	BOOT CFG-2	-
2	0x0A00_B000	0x0A00_BFFF	4 KByte	RSVD	No access.
2	0x0A00_C000	0x0A00_CFFF	4 KByte	RSVD	No access.
2	0x0A00_D000	0x0A00_DFFF	4 KByte	RSVD	No access.
2	0x0A00_E000	0x0A00_EFFF	4 KByte	RSVD	No access.
2	0x0A00_F000	0x0A00_FFFF	4 KByte	RSVD	No access.

Note: All RSVD addresses are “Address Holes” and therefore generate a bus error back to the initiator.

31.1.4 FCW Write Flow

31.1.4.1 Word Write Flow

The address for erases and the address/data for word writes is stored by the CPU in the Special Function Registers (SFRs) of the FCW. The CPU stores this data using Peripheral Bus A. Flash operations are executing using the FCR (Flash Controller, Read).

31.1.4.2 Row Write Flow

Row writes of Flash do not use data already stored in the FCW's registers. Instead the data is read directly from data RAM using the AHB bus matrix.

31.1.5 Flash Retention and Endurance

Flash panels have retention and endurance values native to the process technology in which they are designed.

Two mechanisms exist to increase the endurance while maintaining retention, ECC and Pre-Program.

- ECC masks native error rates by detection and correction. Since ECC can correct one error per Flash Word, it can double the endurance of a panel.

- Pre-Program reduces stress when programming bit cells thereby doubling endurance. However, Pre-Program adds about 20 percent to the flash programming time.
- Using both ECC and Pre-Program can yield a significant endurance improvement.

For additional information, refer to the in the Electrical Specifications section.

31.1.6 ECC Support

The FCW always writes ECC values for Double Word, Quad Word, and Row Write operations. The FCW always writes Parity for Single Write operations. The Flash ECC settings (in FCR ECCCTRL) determine if Single Write operations are allowed. Single Write operations are NOT allowed when Flash ECC settings are set to ECC Writes with ECC Reads. For all Dynamic Write ECC selections, the FCW allows Single Write operations.

Note that the Width of the Flash determines if Single, Double or Quad word programming is supported. See the Control Operation Register (CTRL0P) for flash operations supported for each flash bit width.

31.1.7 Debug Access Level (DAL)

The FCW with respect to the DAL works across various systems.

31.1.8 SFR Write Protect Features

A number of mechanisms exist within the device to ensure that inadvertent writes to program Flash do not occur. They are :

1. Module write protection.
2. An unlock value (KEY) to allowing writing critical registers.
3. Local SFR bits that provide write locks for other SFR bits.
4. SFR bits that provide write/erase protection of Flash memory regions.

Only items 1, 2, and 3 will be discussed here.

31.1.8.1 SFR Write Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC) or the WPCTRL. Write protection does not extend to HSM if present or external debugger accesses. HSM SFRs are protected with HSMWPCTRL and are only accessible by an external debugger connected to the HSM CPU.

See the Control Operation (CTRL0P) register.

31.1.8.2 SFR Write Unlock Values

The KEY register provides write protection to critical NVM registers. KEY uses multiple KEY Values to unlock access to classes of registers. Writing the specific KEY Value to the KEY register unlocks access to the defined set of registers. When the KEY Value protects a multitude of registers the unlock remains in effect until SW clears it. When the KEY Value protects a single register, the KEY clears after a successful write.

KEY is always cleared by a write to a KEY protected register that is currently locked. This causes the INTFLAG.KEYERR bit to be set. SW can clear the unlock state by writing any invalid key to it (recommend all 0's). The INTFLAG.KEYERR is not set by this case.

See the Control Operation (CTRL0P) register for a summary of registers and which ones require a KEY Value to unlock them.

31.1.8.3 Local Lock Bits

Some SFR bits have associated local lock bits. These bits lock updates of the associated feature until the next reset. Lock bits are found in SWAP, PWP, LBWP, UBWP, UOWP, CWP.

31.1.9 FCW Sequencer User Model

31.1.9.1 Sequencer Enable Control

.KEY must be written with the WRKEY value before CTRL0P can be written to start a Write/Erase sequence. Once CTRL0P is written, the KEY register returns to 0's (locks) and all registers in the FCW are write protected (BUSY = 1) until the operation finishes. Write protecting the registers prevents addresses, data and configuration changing mid sequence.

31.1.9.2 Start Sequencer

Once the NVM system is configured the only registers necessary for programming are CTRL0P, KEY, ADDR, and either DATAn or SRCADDR. Page Erase does not need DATAn or SRCADDR. And other Erases do not need ADDR.

For Write/Erase, the steps are as follows:

1. Setup ADDR and if programming either DATAn (Single/Quad) or SRCADDR (Row Write).
2. Write WRKEY to KEY.
3. Write <Desired NVMOP> to CTRL0P.
4. The FCW generates an interrupt when it clears STATUS.BUSY and sets INTFLAG.DONE.

When CTRL0P is written, the write/erase sequence starts and the CPU is unable to execute from the selected Flash panel for the duration of the sequence.

31.1.10 Flash Write Protect Features

Flash write protect features are for safety not security. They protect against inadvertent erases or updates due to execution of errant code.

In a TrustZone system, the trusted portion of the CPU owns the FCW. The trusted NVM driver is responsible for the security check determining if the calling application has rights to modify the requested address.

31.1.10.1 PFM Write Protect (PWP)

The PWP register provides write/erase protection for a region in PFM using PWPn.BASE and PWPn.SIZE. BASE is the flash physical address of the start of the region to be protected. BASE is aligned to the page size of the flash panel. SIZE is the number of pages to be protected. There are 4 regions that can be protected.

Protection can be enabled or disabled using PWPn.PWPEN bit. However, any address range not defined by PWPn is not protected. When PWP regions with different protections overlap, the one with the more restrictive wins. Because of this, overlapping regions are not useful.

See the [Interrupt Flag Summary Table](#) for the effects of write protect errors on INTFLAG.

31.1.10.1.1 PFM Write Protect Mirror

With dual panel systems PWP can be mirrored from the Lower PFM to the Upper PFM OR from the Upper PFM to the Lower PFM. When PWPn.PWPMIR = 1, PWPn is applied to both Upper and Lower PFM regions. The check simply masks the most significant address bit (BASE, SIZE, ADDR) used in the compare.

Individual mirror bits per PWPn regions allows the lower PFM to be protected while the upper region is not (or vice versa).

This becomes important for TrustZone applications in which both the lower and upper TrustZone regions must remain protected and locked while only the lower untrusted regions are protected.

31.1.10.1.2 PWP Example

For example, to protect a 104KB region starting at 0x0134_7000:

BASE = 0x0134_7000 << 8 = 0x3470_0000

SIZE = 104/4 = 26 = 0x1A = 0x0000_001A

PWPEN = 0x0000_8000

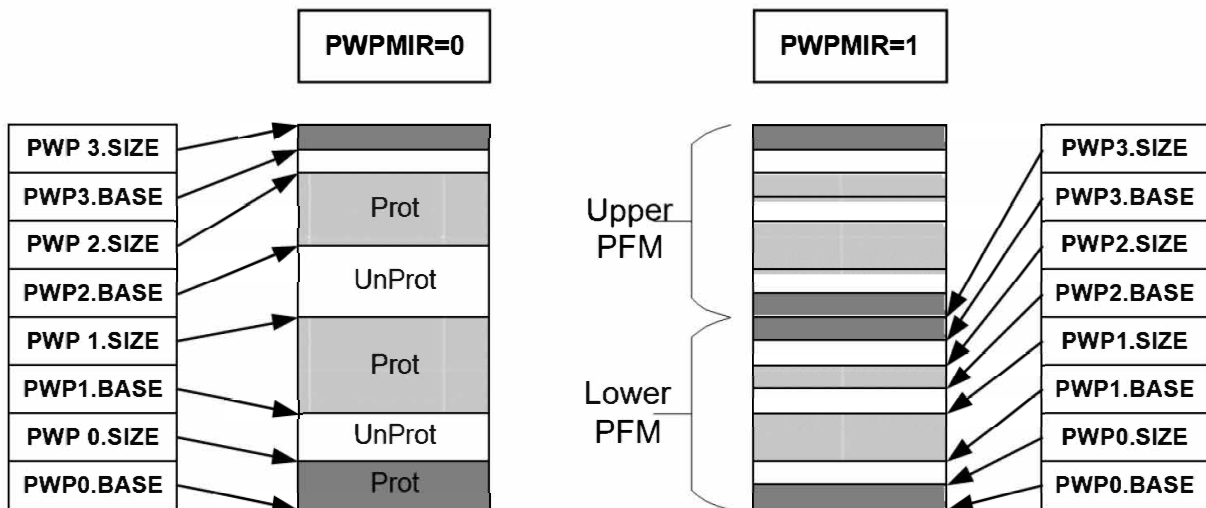
PWPn = BASE | PWPEN | SIZE

The result for PWPn is 0x3470_801A

The following figure shows a graphical representation of Program Write Protect (PWP). Shaded regions are write/erase protected. Non-shaded regions are not protected.

- Values for PWPMIR: Mirror PWP bit
- 1 (MIRROR) = PWP settings are Mirrored.
- 0 (UNMIRROR) = PWP settings are NOT Mirrored, also applies to single panel systems.

Figure 31-2. PWP Example: No Mirror versus Mirror



31.1.10.1.3 PWPx Lock/Unlock Sequence

PWPx registers require a lock/unlock sequence as described in [SFR Write Unlock Values](#). Any attempt to write to the PWPx register when locked has no effect.

31.1.10.1.4 Local Lock Bits

The user may optionally select to prevent further writes to a PWPn register by also setting the local LOCK bit (in the same register) when writing the PWPn values.

When the local lock bit is set (LOCK = 1), subsequent writes to that register (even if the register unlock sequence is followed) have no effect, creating a “write once” register. Local LOCK bits revert to the unlocked state at reset.

31.1.10.2 Boot Write Protect

Note: This section uses BWP to refer to both LBWP and UBWP.

Write protection also includes erase protection

The BWP registers are used to write/erase protect individual pages within BFM. The protect bits are grouped into Lower Boot (LBWPn) and Upper Boot (UBWPn). These are two equal size regions containing the same number of pages. BWP always uses the logical page location which depends on BFSWAP to control protection for the physical page.

At reset, all BWP bits are set to logic ‘1’, write protecting all user accessible BFM pages. Clearing a BFM page bit within its register removes write protect from the corresponding page.

See the [Interrupt Flag Summary Table](#) for the effects of write protect errors on INTFLAG.

31.1.10.2.1 BWP Lock/Unlock Sequence

LBWP and UBWP require a lock/unlock sequence as described in [SFR Write Unlock Values](#). Any attempt to write to these registers when locked will have no effect.

31.1.10.2.2 Local Lock Bits

The user may optionally select to prevent further writes to a BWP register by also setting the local LOCK bit (in the same register) when writing the BWP value.

When the local lock bit is set (LOCK = 1), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a “write once” register. The local LOCK bit reverts to the unlocked state at reset.

See Lower (LBWP) and Upper (UBWP) BFM Write Protect Registers for all possible settings regarding the lock and protection bits.

31.1.10.3 Configuration Write Protect

The CWP register is used to write/erase protect individual pages within CFM. A single register supports the CFM for a 1 Panel system, and Panels 1 and 2 for the dual flash model.

At reset, all CWP bits are set to logic ‘1’, write protecting all user accessible CFM pages. Clearing a CFM page bit within its register removes write protect from the corresponding page.

The CWP register contains write/erase protection control for Configuration Flash Memory (CFM).

All VSS pages are reserved for use by the HSM if the device family supports one. Some pages are always protected from write/erase commands as noted in the access type for the CWP bit.

31.1.10.3.1 CWP Lock/Unlock Sequence

This register requires a lock/unlock sequence as described in [SFR Write Unlock Values](#). Any attempt to write to this register when locked will have no effect.

31.1.10.3.2 Local Lock Bits

The user may optionally select to prevent further writes to a CWP register bit by also setting the local LOCK bit (in the same register) when writing the CWP value.

When the local lock bit is set (LOCK = 1), subsequent writes to that register bit (even if the unlock sequence is followed) have no effect, creating a “write once” register bit. Local LOCK bits revert to the unlocked state at reset.

Some bits cannot be modified but show the protection state of the CFM page. VSS pages cannot be modified by the system for devices that have an HSM option and are shown as Read Only.

See CWP Register and HSMCWP Register for full interaction of lock and protection bits.

31.1.10.4 User-OTP Write Protect

One page in the flash boot region (of each panel in a dual panel system) is implemented as One-Time Programmable NVM called UserOTP. It can only be written and cannot be erased (by page or chip erase). It is intended for the storage of user system calibration data that must survive a flash erase.

UserOTP can be used to store and preserve identification values such as an Ethernet MAC addresses, OEM/ODM version numbers, board version numbers, etc. Another use would be for calibration values of circuits external to the chip but constant for a board design in which the chip exists.

The UOWP register is used to write protect 1/4 sized regions of the UserOTP page(s). UOWP.UO1WP0 write protect the lowest address quarter page in Panel 1 while UOWP.UO2WP3 write protects the highest addressed quarter page in panel 2.

31.1.10.4.1 UOWP Lock/Unlock Sequence

The UOWP register requires a lock/unlock sequence as described in [SFR Write Unlock Values](#). Any attempt to write to this register when locked has no effect.

31.1.10.4.2 Local Lock Bits

The user may optionally select to prevent further writes to a UOWP region bits by also setting local LOCK bits (in the same register) when writing the UOWP value. There are individual lock bits per ¼ region.

When a local lock bit is set (LOCK = 1), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a “write once” register. Local LOCK bits revert to the unlocked state at reset.

31.1.11 Write Sequences

Reads from the panel being programmed stall until the programming sequence finishes. This is because FCW programming requests have the higher priority than read requests.

Note: Performing any programming sequence when power is lost results in an undefined increased wear of the panel. Loss of power is defined as receiving a POR event.

31.1.11.1 Pre-Program

The Flash panels in this device support an option to Flash programming that increase endurance of the panel. This feature is called Pre-Program. When selected, the controller automatically initiates two programming cycles per write. The first is a short cycle and the second is a full cycle. This option adds about 20% to the programming time.

To use this feature set the CTRL0P.PREPG SFR bit when writing CTRL0P.NVMOP. In all other respects the SFR setup is identical. Pre-Program typically doubles the native Endurance and Retention of the panel. It is recommend to either always use or never use Pre-Program feature. Mixed use of PREPG yields undefined endurance and retention of the panel.

It is highly recommended to always have CTRL0P.PREPG = 1.

31.1.11.2 Single Write Sequence

The smallest block of data that can be programmed with a single write either one Word or one DWord. The data to be programmed must be written to DATA0 (and DATA1 for DWord) before the programming sequence is initiated. The FCW programs the DATA_n into the Word/DWord location pointed to by ADDR. ADDR is a byte address, so the lower 2 or 3 bits are ignored for Word or DWord programming, respectively.

A Word/DWord Program sequence typically comprises the following steps:

1. <Desired NVMOP> is Single Write.
2. Follow the Start Sequencer from [Start Sequencer](#).
3. Wait for NVM Interrupt from [Interrupts](#).
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See [Errors and Flags](#) about error flags.

A Word/DWord within the Flash can be programmed if its associated page write protection is not enabled, see [Debug Access Level](#).

31.1.11.2.1 Single Write Timing

Word/DWord Program timing is dominated by setup (T_{nv}s + T_{pg}s), program time (T_{prog}) and recovery (T_{rcv}) delays. Using the timing shown in “Flash Timing Parameters”, total time to program one word is roughly shown below:

Word/DWord Program time = T_{nv}s + T_{pg}s + T_{prog} + T_{rcv} + T_{rw}

If Pre-Programming, the first step adds roughly:

Word/DWord Program time = T_{preprog} + T_{prepg} + T_{prepgs}

Note: For write timing, see the Non-Volatile Memory Controller (NVM) Electrical Specifications section in the Electrical Characteristics chapter.

31.1.11.2.2 Single Write and ECC

Single Writes are not allowed because of Flash ECC setting. The [Interrupt Error Flag Summary](#) table defines the interrupt flags that are set.

31.1.11.3 Quad Write Sequence

The Quad Write operation implements a Quad Double Word programming. The 4, 32-bit instruction/data words to be programmed must be written to $DATA_n$, $n = 0, 1, \dots, 3$, before the programming sequence is initiated. The Flash instruction word at the location pointed to by ADDR is programmed. The program operation must be aligned to a Flash word address, so the lower 5 bits are ignored for Quad Write.

The Quad Write sequence typically comprises the following steps

1. <Desired NVMOP> is Quad Write.
2. Follow the Start Sequencer from [Start Sequencer](#).
3. Wait for NVM Interrupt from [Interrupts](#).
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See [Errors and Flags](#) about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to '0' and the OWNER field to '00' simultaneously to the MUTEX register.

A flash address can be written if its associated page write protection is not enabled, see [Debug Accesss Level](#). The FCW writes whatever is in $DATA_n$ when commanded.

31.1.11.3.1 Quad Write Timing

Quad Write timing is dominated by setup ($T_{nvs} + T_{pgs}$), program time (T_{prog}) and recovery (T_{rcv}) delays. Using the specs shown in Non-Volatile Memory Controller (NVM) Electrical Specifications, the total time to program one QWord/QDWord is roughly:

$$\text{Quad Write Time} = T_{nvs} + T_{pgs} + 4 \cdot T_{prog} + T_{rcv} + T_{rw}$$

If Pre-Programming, the first step adds roughly:

$$\text{Quad Write Time} = 4 \cdot T_{preprog} + T_{prepgh} + T_{prepgs}$$

Quad Write asserts programming voltage throughout the sequence to avoid duplicating setup and recovery delays. Consequently, a Quad Write operation takes less time than 4 Word/DWord Program operations.

31.1.11.3.2 Quad Write and ECC

Note: All configuration data held in flash and read before user boot up must be programmed, by software, using Quad Write or it will generate an ECC error when read.

31.1.11.4 Row Write Sequence

The largest block of data that can be programmed by a single NVMOP command is a row. As shown in Flash Module Construction, a row is 1024 bytes of data. ADDR is the row aligned address where the Flash address starts programming the data. The controller ignores the sub-row address bits and ALWAYS starts programming at the beginning of a row.

A Row Write sequence comprises the following steps:

1. Write the entire row of data to be programmed into system SRAM. The source address must be word aligned and secure/non-secure zone consistent with the transaction type.
2. <Desired NVMOP> is Row Write.
3. Follow the Start Sequencer from [Start Sequencer](#).
4. Wait for NVM Interrupt from [Interrupts](#).
5. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See [Errors and Flags](#) regarding error flags.

A row of Flash can be programmed if its associated page is not write protection see *Debug Access Level*.

The data transferred from RAM is buffered within the FCW. The FCW fetches enough data words to calculating ECC for each programming operation in the row.

The FCW automates programming the data into the Flash using contiguous Single Write operations. The data is double buffered, such that each programming sequence of four programming operations may execute while the next data are read from the internal system SRAM.

Notes:

1. The base address held in the SRCADDR register is not changed during the Row Write sequence.
2. The DATAn registers may be affected by the Row Write sequence and cannot be guaranteed to hold prior data.

31.1.11.4.1 Row Write and ECC

Since Row Write uses multiple Quad Writes to update a row, it has the same features and restrictions as Quad Write.

Note: All configuration data held in flash and read before user boot up must be programmed, by software, using Quad Write or it will generate an ECC error when read.

31.1.12 Erase Sequences

Note:

Any access to the panel containing the page or panel being erased stalls reads from all panels until the erase finishes.

31.1.12.1 Page Erase Sequence

A page erase performs an erase of a single page of PFM or BFM, or select pages in CFM (User, Boot), as long as it is not write protected. . The page to be erased is selected using ADDR. The lower bits of the address given by ADDR are ignored in page selection.

Note:

- 1) During the page erase, accessing any uncached/unbuffered address within the panel containing the page causes that access to stall until the erase finishes. It is user code responsibility to prevent or handle that situation.
- 2) It is also the responsibility of the programmer's code to ensure that no bus initiators, including the CPU, access the Flash region, the entire PFM of the panel, or the entire Boot of the panel, that is to be - or is being - erased.

31.1.12.1.1 Page Erase

A Page Erase sequence comprises the following steps:

1. <Desired NVMOP> is Page Erase.
2. Follow the Start Sequencer from [Start Sequencer](#).
3. Wait for NVM Interrupt from [Interrupts](#).
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See [Errors and Flags](#) about error flags.

A page of Flash can be erased if its associated page write protection is not enabled, see [Debug Access Level](#).

31.1.12.1.2 Page Erase Timing

Page Erase timing is dominated by setup (T_{nvs}), erase time (T_{erase}) and recovery (T_{rcv}) delays. Using the timing shown in Non-Volatile Memory Controller (NVM) Electrical Specifications, the total time to erase one page is:

$$\text{Time to Erase Page} = T_{nvs} + T_{\text{erase}} + T_{rcv} + T_{rw}$$

31.1.12.1.3 Page Erase Retry

Erase Retry is not longer needed to achieve 20K cycle endurance, use Pre-Program for 20K.

31.1.12.2 PFM Single or Dual Panel Erase

Note: Upper and Lower PFM erase is not applicable to Single Panel devices.

For a dual panel system, the Upper PFM erase NVMOP command erases the PFM in the panel mapped to the Upper PFM address range. Lower PFM erase NVMOP command erases the PFM in the panel mapped to the Lower PFM address range. These commands leave the BFM and CFM pages intact, and are intended to be used by a field upgradeable device. To guard against accidental erasure user code can setup PWP for protection.

The ability to erase the PFM in the Upper or Lower region allows executing from one panel while erasing (or writing) the other, without the need to complete many Page Erase operations.

The PFM Erase operation is not address based (i.e. the FCW directly selects the target flash panel). The ADDR register is ignored so does not need to be initialized for this NVMOP.

A Panel Erase sequence comprises the following steps:

1. <Desired NVMOP> is Upper or Lower PFM Erase.
2. Follow the Start Sequencer from [Start Sequencer](#).
3. Wait for NVM Interrupt from [Interrupts](#).
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See [Errors and Flags](#) about error flags.

The Flash memory in a panel mapped to the Upper or Lower PFM region can be erased if write protection is not enabled for addresses associated with that region, see [Debug Access Level](#).

If in mission mode, the application must NOT be executing from the erased region. If any initiator reads from the panel containing the region being erased, all Flash accesses are stalled until the erase completes.

31.1.12.2.1 Program Erase Timing

Program Erase timing is dominated by setup (Tnvs), erase time (Terase) and recovery (Trcv) delays.

Program Erase = 2 * (Tnvs + Terase + Trcv + Trw)

Note: For write timing, see the Non-Volatile Memory Controller (NVM) Electrical Specifications section in the *Electrical Specifications* chapter.

31.1.12.3 PFM All Panel Erase

The PFM Erase NVMOP command erases the entire PFM region. This includes the PFM region in both panels of a dual panel system. This mode leaves the BFM and CFM pages intact and are intended to be used by a field upgradeable device. To guard against accidental erasure user code can setup PWP for protection.

A Single or Dual Panel Erase sequence comprises the following steps:

1. <Desired NVMOP> is both Upper and Lower PFM Erase.
2. Follow the Start Sequencer from [Start Sequencer](#).
3. Wait for NVM Interrupt from [Interrupts](#).
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See [Errors and Flags](#) about error flags.

All PFM panels can be erased only if all associated page write protection is not enabled, see [Debug Access Level](#).

If in mission mode, the application must NOT be executing from the erased region. If any initiator reads from the panel containing the region being erased, all Flash accesses are stalled until the erase completes.

31.1.13 Boot Support

The NVM Controller supports the following boot methods:

- Dual Boot regions each with half the available total BFM size.
- Single contiguous Boot region using the total BFM size.

Additionally, a Secure Boot page can be setup within each Boot region in both Single and Dual Boot modes.

31.1.13.1 Single and Dual Boot

In a dual panel system, each panel has BFM. The BFM can be used in either Dual Boot or Single Boot. Dual Boot designates each panel's BFM as a boot source. This allows the safe updating of one boot image while the other image stays intact. Single boot allows the boot image to span both panel's in the single BFM.

For BLM you can select to lock a swap setting or determine how a panel is swapped. However, you cannot choose if a single or dual boot will be initiated. The Microcontroller makes this decision based on how your device is designed.

31.1.14 BFM and PFM Panel Swapping

For BFM and PFM you can select to lock a swap setting or determine how a panel is swapped. The following register values are used to determine how panels are swapped:

- **BFM**
SWAP.BFSWAP, 1 = Upper and Lower BFM Regions are Swapped.
SWAP.BFSLOCK, 1 = PFSLOCK and PFSWAP cannot be written.
- **PFM**
SWAP.PFSWAP, 1 = Upper and Lower PFM Regions are Swapped.
SWAP.PFSLOCK, 1 = PFSLOCK and PFSWAP cannot be written.

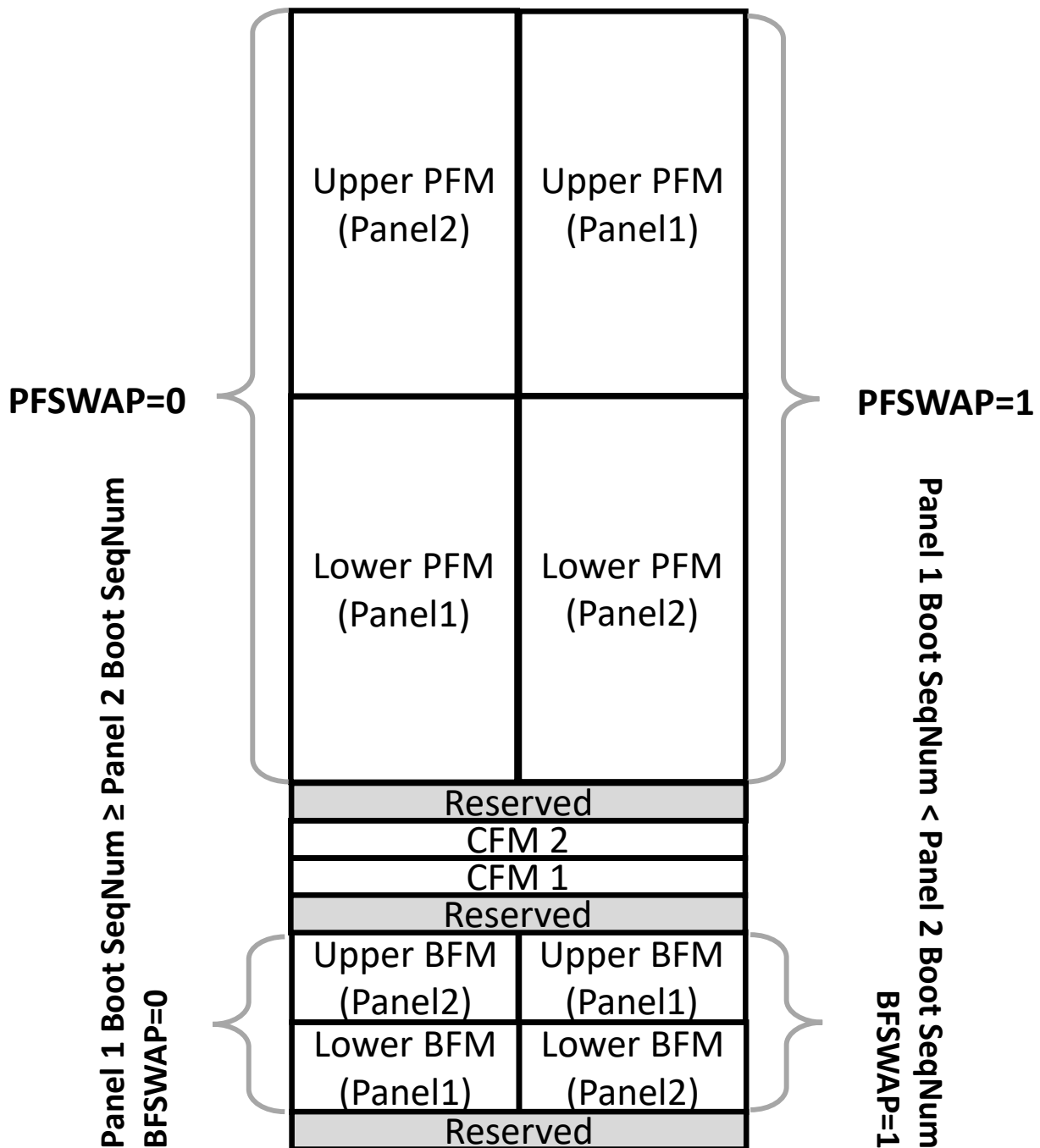
An explanation of these settings follows.

31.1.14.1 Manual Panel Swap

The memory region order of the BFM (and PFM) panels is entirely the user's responsibility. The control bit, SWAP.BFSWAP (SWAP.PFSWAP), provides the manual control to swap the order of the panels. In order to properly swap the BFM (or PFM) panel regions, there must not be any accesses to either panel when changing the BFSWAP (PFSWAP) bit. Accesses to regions not being swapped, or SRAM is allowed.

An example device memory map is shown in the following figure with the PFSWAP bit set or cleared.

Figure 31-3. Device Memory Map Example



31.1.14.2 BFM and PFM Swap Organization Notes

Single Panel Device Organization Notes

When a device contains a single panel flash system, swapping changes how logical system addresses are mapped to physical flash addresses within the single panel. When not swapped, the Lower region maps to the lower half of the physical flash addresses and the Upper region maps to the upper half of the physical flash addresses. When swapped, the Lower region maps to the upper half

of the physical flash addresses and the Upper region maps to the lower half of the physical flash addresses.

Dual Panel Device Organization Notes

When the devices contains a Dual Panel flash system, swapping changes the panel in which the Upper and Lower regions exists. The natural order, not swapped, is for the Lower region to be mapped to the lower numbered panel and the Upper region to be mapped to the higher numbered panel. When swapped, the Lower region is mapped to the higher number panel and the Upper region is mapped to the lowered number panel.

Since a dual panel system supports Live Update (aka Read While Write - RWW) for access to different panels, swapping BFM when not swapping PFM can impact code execution if using writes and erase NVMOP operations during mission mode execution.

BFM Swap Protection

It may be desirable to prevent swapping BFM after setup. The BFSLOCK allows BFSWAP (and itself) to be write locked preventing them from changing. Lock is in effect until the next reset.

PFM Swap Protection

It may be desirable to prevent swapping PFM after setup. The PFSLOCK allows PFSWAP (and itself) to be write locked preventing them from changing. Lock is in effect until the next reset.

31.1.15 Operation Timing

The FCW supports only self-timed operations using an internal clock source. The timing values for writing and erasing the Flash are predefined. No user involvement is need for timing value selection.

The AHB bus host initiator operates independently of the FSM, transferring data to the FCW data buffer only when requested. Consequently, there are no timing restrictions with respect to AHB and the FSM. However, for Row Write operations, the AHB interface must have sufficient bandwidth to SRAM to keep up and therefore not starve the FCW for data.

31.1.16 Interrupts

The Flash module has an interrupt output that is controlled by INTENSET and INTENCLR. The interrupt status is reported via INTFLAG. A read of either INTENSET or INTENCLR returns the enable state for the specific interrupt condition. The Flash interrupt is asserted if a condition is enabled and its associated bit is set in INTFLAG. Interrupt flags are cleared by writing a 1 to its bit position in the INTFLAG register.

It is recommended to always enable DONE as interrupt source in INTENSET. Using DONE masks side effects of WRERR, RSTERR, BUSERR, and FIFOERR being set before the FCW has finished cleaning up after an error.

The NVM Busy Status (STATUS.BUSY) indicates if the FCW is performing an operation (from the CPU). When NVMOP completes, the FSM clears BUSY and sets Done.

The NVM Operation Done Flag (INTFLAG.DONE) indicates that the FCW has completed an NVMOP (for the CPU).

Note: The FCW does not clear DONE automatically. Also, DONE = 1 does not prevent NVM operations. Therefore, software must clear DONE before the next NVM operation is started.

31.1.17 Errors and Flags

Several non-typical events can occur that affect the operation of the FCW causing Write/Erase operations to be aborted. Setup errors can occur that prevent the start of an NVMOP command. All these errors are reported via INTFLAG. Some flags are updated before the FSM clears BUSY. However, the FSM always clears BUSY and sets DONE on the same clock edge.

The following table describes the types of errors the FCW detects and what flags they affect.

Table 31-3. Interrupt Flag Summary

INTFLAGS:	Source:	Effect on FCW Operations:
WRERR, DONE	Any Error Event during NVMOP	Last address programmed may be corrupted, or last erase may not be complete.
RSTERR, DONE	BOR circuit not Ready before NVMOP written	NVMOP not started ⁽²⁾ SW should enable BOR and wait for it to be ready.
RSTERR ⁽¹⁾	Reset or Low Voltage Event before FSM starts NVMOP	NVMOP not started ⁽²⁾
RSTERR ⁽¹⁾ , WRERR	Reset or Low Voltage Event after FSM starts NVMOP	Last address programmed may be corrupted, or last erase may not be complete. Software should verify results.
HTDPGM, DONE	High Temp Event before FSM starts NVMOP	NVMOP not started ⁽²⁾
WRERR, SECERR, DONE	Security module or tamper event preempted operation	Last address programmed may be corrupted, or last erase may not be complete.
KEYERR	KEY not valid for SFR write	SFR not updated
CFGERR, DONE	ADDR not in flash memory space Disallowed CPUDAL setting.	NVMOP is not started ⁽²⁾
CFGERR, OPERR, DONE	NVMOP not allowed because of ECC, i.e. Single Write attempted when ECC Write with ECC Read is selected	NVMOP is not started ⁽²⁾
WRERR, FIFOERR, DONE	Row Write FIFO under run error	NVMOP is aborted ⁽²⁾ Subsequent addresses not programmed
BUSERR, FIFOERR ⁽³⁾ , DONE	AHB Bus Host error before FIFO filled Possible SRCADDR out of range	NVMOP is not started ⁽²⁾
BUSERR, FIFOERR ⁽³⁾ , WRERR, DONE	AHB Bus Host error Possible SRCADDR out of range or access permission violation	NVMOP is aborted ⁽²⁾ Current and subsequent addresses not programmed
WPERR, DONE	ADDR is write protected by a WP register (i.e BWP, PWP, etc)	NVMOP is not started ⁽²⁾
WPERR, SECERR, DONE	ADDR is write protected by a different security association ⁽⁴⁾	NVMOP is not started ⁽²⁾
OPERR, DONE	NVMOP is not available (e.g. RSVD)	NVMOP is not started ⁽²⁾
SECERR	Security Violation	SFR access blocked Does not generate a bus error

Notes:

1. If BOR causes a reset, DONE will not be set because it is cleared by the reset.
2. NVMOP is generally considered started when the panel is enabled.
3. Error code may or may not be set.

The following table summarizes events that cause the Write/Erase sequence to not occur but do not set the error flag.

Table 31-4. Conditions Without an Error Flag

Source:	Effect on FCW Operations:
Writing KEY with an invalid value.	Locks all registers protected by KEY.
Attempted write of an SFR while locked for programming (BUSY=1) from the same security association.	Action is ignored by FSM and SFRs. Reports a bus error.
Attempted read or write of an SFR address that is not implemented.	Action is ignored by FSM and SFRs. Reports a bus error.

31.1.17.1 Write Error (INTFLAG.WRERR)

The FCW Finite State Machine (FSM) set the WRERR flag only if the FSM aborts a Write/Erase operation (i.e. an NVMOP command). The FSM sets the flag on the detection of an error which

is before it clears BUSY and sets DONE. WRERR is only set if there is possible corruption of the flash due to the NVMOP abort.

31.1.17.2 Reset and Low Voltage (BOR) Error (INTFLAG.RSTERR)

If a reset or BOR event occurs while an NVMOP command is underway, the FCW sets flags as per the [Interrupt Flag Summary](#) table and aborts row programming (on a quad write boundary) and all erases. A system reset prevents an interrupt from occurring. However, relevant INTFLAG bits are reset on POR and maintain state through a system reset and a BOR. Therefore, software can check the prior state of the FCW. It can also check the device's RCAUSE register to see if the error is from a system reset or from a BOR. If the system shows a POR occurred, software has no guaranteed way of knowing if the system had been operational and a write/erase operation was underway at the time the POR occurred.

31.1.17.3 Power-on Reset

A Power-on Reset (POR) causes immediate termination of any FCW operation. A POR is a full and hard reset of all states of the FCW and therefore no error flags originating from a prior operation survive the reset.

Several SFR bits get reset by POR so they survive a BOR or user resets (WDT Reset, RESET pin, etc). Doing so allows many NVM operations to complete and the ones that don't can report errors. Also, they prevent values like address from changing in the middle of an operation.

31.2 Flash Controller, Read

31.2.1 Overview

The Flash Read Controller provides up to 4 AHB system read targets for the flash, an ECC calculation module, automatic CRC scanning of Flash, and Fault Injection. It also provides arbitration between the FCW and the system, ready timers for startup from POR/sleep, and multiple panel address decoding.

The FCR provides a user interfaces to Flash read control features. The FCR contains Flash wait state control, ECC fault reporting, ECC Fault Injection, and Flash CRC. The Fault and CRC features that can be used to meet certain functional safety requirements for invariant memories.

For additional information, refer to the [Definitions](#) and [Flash Architecture](#).

31.2.2 Features

The following are key features of the FCR module:

- Supports Flash partitions
 - Program Flash Memory (PFM)
 - Boot Flash Memory (BFM)
 - Configuration Flash Memory (CFM)
- Flash-Based Configuration Retention Support:
 - Read and update external module Calibration and Configuration data
 - Read and update Flash IP Calibration data
 - Read and update Page Redundancy Records (locally)
- Flash Memory Read Protection for pages in CFM
 - ROM CFG and Boot CFG x
 - VSSn (Secure Storage pages)
- AHB Reads of Flash Memory
 - Up to 4 AHB Read ports
 - One Read Buffer per port

- CRC Logic for In-variant memory tests
- Fault Logic for error injection and testing

31.2.3 Concurrent Access

31.2.3.1 Read While Write

In order to permit code execution from one flash panel while writing (or erasing) another, the Flash Read Controller (FCR) macro must be able to detect and arbitrate access conflicts between the Flash Write Controller (FCW), System read requests, and internal read requests.

When a panel can no longer be read because it is actively being written or erased, the panel arbiter is signaled accordingly. Read accesses from the system bus are compared to the panel address ranges and allocated accordingly within the panel arbiter.

31.2.3.2 Concurrent Reads

The FCR's Arbiter permits system reads to access any panel, provided it is not being actively written or erased. The panel arbiter also permits concurrent read accesses to different panels.

If the system (CPU or other peripheral initiator) attempts to access a panel that is being actively written or erased, then the FCR's Arbiter delays the data ready response to the initiator until such time that the FCW has completed its sequence (i.e. the FCW always has priority).

31.2.4 Flash Address Map Swap

31.2.4.1 PFM Address Map Swap

The FCR handles PFM address map swapping for read operations based on the value of SWAP.PFSWAP in the FCW. See the SWAP Register for more information on PFSWAP.

31.2.4.2 BFM Address Map Swap

The FCR handles BFM address map swapping for read operations based on the value of SWAP.BFSWAP in the FCW. See the SWAP Register for more information on BFSWAP.

31.2.4.3 CFM Address Map Swap

The Configuration Flash Memory address map cannot be swapped.

31.2.5 Power Management

The Flash system supports multiple power modes for reducing or managing power. These include Auto-Standby, Flash power down, high temperature operation, and controlling power during sleep.

31.2.5.1 Flash Auto Standby

The Flash panel enters a standby mode automatically after a read or write operation. Each Flash panel enters its Auto-Standby independently of the other. After a read request completes as defined by the read access time (T_{ACC}), the panel enters Auto-Standby. For writes/erases this after the recovery time. When a panel is in this mode read and write requests incur no access time penalty.

31.2.5.2 Flash Hibernate

The Flash panel supports Flash Hibernate mode which removes power from most of panels circuits. This substantially reduces the Flash systems power consumption. Flash Hibernate affects both panels.

Flash Hibernate incurs an entry and exit delay, T_{DPDS} and T_{DPDH} , respectively. When in Flash Hibernate all Flash operations are suspended, however, the Flash control system automatically wakes up the panels on a request (read or write). The panels are not automatically put back into hibernate after a wakeup event. If the interrupt causes a wakeup to sleepwalking the panels return the mode selected by CTRLB.SLP.

See *Power Mode in Standby Sleep* for manual entry and exit of Flash Hibernate.

31.2.5.3 Sleep Interaction

The Flash system holds off entry into sleep modes lower than Idle Sleep to cleanly stop Flash operations. The following operations delay entry into sleep:

- In-progress read from any source (system, CRC, etc)
- Flash mode change (i.e. High Temp, wake from Flash Hibernate)
- Requested program or erase operation, etc.

Note: Row programming and erase operations can significantly delay entry into a sleep mode.

Entry into Idle Sleep mode does not cause a flash mode change. Clocks in the Flash system stop automatically when they are not needed for operation.

Entry into Standby Sleep mode can cause a Flash mode change. The flash can be setup to enter Auto-Standby or Flash Hibernate. If Flash Hibernate is selected there is the option to wakeup to Auto-Standby or remain in Flash Hibernate until the first access. Waking up to Auto-Standby provides the fastest initial access to the Flash system.

See *Power Mode in Standby Sleep* for power mode control of the Flash on entry and exit of system Standby Sleep.

31.2.6 Flash Read Control

31.2.6.1 Read Control Overview

The FCR provides read access to the Flash system for the CPU and peripheral hosts. It implements 4 AHB client. This interface runs at the same frequency as the system bus and uses the same Flash access timing control.

31.2.6.2 CTRLA SFR Description

The CTRLA fields can be changed at anytime, provided the FCR is not write protected. New wait state values are used on the next flash read. RDBUFWS changes are used on the next update to the associated read buffer. A new arbitration value is used at the next available arbitration cycle on a per panel basis.

The Register values for CTRLA are discussed below.

31.2.6.2.1 ARB - AHB Arbitration scheme (conditional)

ARB controls read requester arbitration to the Flash. Requesters include each AHB interface (numbered 0 to 3) and the CRC Finite State Machine. The main CPU always uses AHB0 to access the Flash. FCW requests take precedence over read requests. Arbitration is per Flash panel.

When ARB = 0, round-robin arbitration is used for all read requests to the Flash. When ARB = 1, fixed arbitration is used for all reads to the Flash. In this mode, AHB0 has the highest priority followed by AHB1, etc., and CRC has lowest priority.

31.2.6.2.2 RDBUFWS[3:0] - Read Buffer Wait States

The RDBUFWS field controls the determinism of accesses from the read buffer. Each bit corresponds to an AHB Interface n , $n = 0$ to 3. When $RDBUFWS[n] = 0$, the associated AHB interface returns data from the read buffer in zero wait states. When $RDBUFWS[n] = 1$, the associated AHB interface returns data with wait states to match the flash access of $ADRWS + FWS$ such that the initiator see a constant access time.

The setting of $RDBUFWS[n]$ is ignored when $AUTOWS = 1$. For this case, all data is returned from the read buffer with zero wait states.

For devices with a flash cache, $RDBUFWS[n]$ applies to its interface only if the cache is disabled.

Accesses to the flash from debug mode are completed via the read buffer.

31.2.6.2.3 AUTOWS - Automatic Wait State Enable

The AUTOWS field controls the use of automatic wait state for the Flash access time (Taws). This is useful when varying clock frequencies.

When AUTOWS=1, the total Flash wait states are ADRWS+Taws. When AUTOWS=0 the total Flash wait states are ADRWS+FWS. Non-CPU ports may incur an extra wait state. Note that the bus may add pipeline delays to either the address phase or the data phase.

31.2.6.2.4 ADRWS - Address Wait State Enable

The ADRWS field controls the number of extra clock cycles for address setup to the Flash memory. The Address wait state occurs before requesting arbitration, so it is in between the bus and the request to Flash. In most systems, the path starts at the initiator of the request.

When ADRWS = 0, no extra cycle clocks are added to address setup. When ADRWS = 1, one extra clock is added to address setup. ADRWS only affects AHB interfaces. It does not affect internal read requests such as from CRC logic if present.

31.2.6.2.5 FWS[3:0] - Flash Access Time Wait States (Defined in terms of AHB Clocks)

The FWS[3:0] field controls the number of extra clock cycles it takes to access the Flash memory and provide data back (to the bus or internal requester). The number of wait states needed is typically Taws/System Clock Period (or AHB Clock Period) rounded up to the nearest integer. If ECC is enabled an extra wait state may be needed. Each device characterizes and states the frequency range for each wait state selection and for ECC on/off. See *FLASH CHARACTERIZATION*.

31.2.6.2.6 PRIV - Privileged Access Only (conditional)

The PRIV field determines if the FCR module accepts all accesses or only privileged accesses. It is set to privileged (1) by default.

31.2.6.3 CTRLB SFR Description

The CTRLB fields can be changed at anytime, provided the FCR is not write protected. Updates that affect the cache happen on read access boundaries.

The Register values for CTRLB are discussed below.

31.2.6.3.1 PREFDIS - Prefetch Disable (conditional)

PREFDIS is only valid when the cache is enabled. When PREFDIS=1, the prefetch buffer is disabled preventing speculative fetches from flash. When PREFDIS=0, the prefetch buffer is enabled allowing speculative fetches from flash.

31.2.6.3.2 CHEDIS - Flash Cache Disable (conditional)

The cache is enabled by default but can be disabled using CTRLB.CHEDIS. The cache is enabled and disabled on access boundaries. When enabled reads obey the PREFDIS setting. When disabled it is invalidated and reads obey the CTRLA.RDBUF[FCR_PCHE_AHB_NUM] setting.

Accesses to the flash from debug mode does not affect the flash cache and its state is not changed.

31.2.6.3.3 SLP[1:0] - Power Mode in Standby Sleep

The SLP field controls the Power Reduction Mode during System Standby, Sleep. Waking the flash from its power down mode takes at least T_{DPDH} . Upon system wakeup, the current Power Reduction Mode is reflected in STATUS.PRM. See the SLP SFR field for more details.

31.2.6.3.4 CHEINV: - Flash Cache Disable (conditional)

The cache is invalidated for any write or erase of the flash, including in either panel of a two panel system. It is also invalidated for a change in SWAP or read protection (via CRP register). Alternately, it can be invalidated manually by writing a 1 to CTRLB.CHEINV.

31.2.6.3.5 TEMP - High Temp Read Control (conditional)

Some Flash panels have a High Temperature Read mode that supports read operations up to 150C junction. High Temp Read is a Read Mode of the Flash and therefore the Flash must be placed in this

mode when reading at high temperatures is required. The High Temp Read mode incurs a longer Flash access time regardless of actual temperature.

CTRLB.TEMP controls entry/exit to/from High Temp Read operation. STATUS.TEMP reflects the current state of High Temp Read. The reset value of STATUS.TEMP is controlled by a Flash-based user configuration fuse typically called NVMTEMP.

Note: If the device can power up at high temp, then the device configuration bit, NVMTEMP, must select High Temp. Boot code can determine the device temperature and update TEMP accordingly.

The TEMP field allows the user to manually control the High Temp Read mode or Standard Read mode. Writing CTRLB.TEMP to a 1 toggles the temperature read mode. When changing to High Temp Read mode, change FWS to support T_{ACC} (HIGH TEMP) before writing to CTRLB.TEMP. When changing from High Temp reads, change FWS after reading STATUS.TEMP = 0. If AUTOWS = 1 then no wait state changes are needed.

Note: It is recommended that if the device can be arbitrarily operated at high temperature, to always have High Temp Reads enabled.

Note that TEMP only affects read operation. Write/Erase operation is controlled an on board temperature monitor.

The following sequence can be performed while code is executing from Flash (or any other memory). If the sequence is performed with STATUS.PRM = 1, once the Flash wakes up it will be in the selected mode.

1. Read current TEMP value: curr_temp = STATUS.TEMP.
2. Write 0x02 to BYTE 0 of CTRLB (toggle Temperature Read mode).
3. Wait until STATUS.TEMP toggles. while (curr_temp == STATUS.TEMP) {}.

No flags get set nor are interrupts generated from a read mode change.

Note: Changing TEMP changes the read mode of the Flash. Flash read mode changes stall Flash reads for TMS + TMH and in addition to synchronization delays.

31.2.6.3.6 PRM - Power Reduction Mode

The PRM field allows the user to manually control the Flash power mode. Writing CTRLB.PRM to a 1 toggles the power reduction mode. The current mode is reflected in STATUS.PRM. When in Flash Hibernate the panel wakes up on access but does not return to Flash Hibernate. Note that when awake the Flash reenters Auto Standby if not accessed for one read cycle time.

To change the Flash Power Reduction Mode:

1. Read current PRM value. curr_prm = STATUS.PRM.
2. Write 0x01 to BYTE 0 of CTRLB (toggle Flash Power mode).
3. Wait until STATUS.PRM toggles. while (curr_prm == STATUS.PRM) {}.

No flags get set nor are interrupts generated from a power reduction mode change. The power status of the Flash is reflected in STATUS.PRM.

Note:

Changing the Power Reduction Mode incurs a delay of TDPDH to take effect.

31.2.7 Cyclic Redundancy Check (CRC) Support

31.2.7.1 CRC Overview

This feature is conditional and may, or may not, be available in your device.

The flash system supports automated CRC of any contiguous flash region. It supports both 16-bit and 32-bit CRC with a programmable polynomial, initial value, final XOR value, and checksum compare value. The CRC can run on-command or continuously with auto repeat.

All control registers must be setup prior to enabling CRC operation. Once setup, writing $CRCEN = 1$ loads the Initial Value into the Accumulator, the Message Length into the Message Length counter, and the Period into the Period counter. After the Period counter decrements to zero the logic reads the data at the Message Address. For each byte shifted through the LFSR, it decrements the byte counter.

Bytes shift into the LFSR lowest addressed to highest addressed. The Reflected Bit Order field, $REFIN$, determines if it is MSbit first or LSbit first. During the shifting the Accumulator bits toggle showing the result. Once the state machine completely shifts in the last byte from the current flash read, it reloads the Period counter and starts counting.

The CRC logic continues reading, shifting, and waiting until the Message Length counter reaches zero. The final step of the CRC operation XORs the Accumulator with the Final XOR value before comparison to the Check Sum value. Reflected Bit Order field, $REFOUT$, determine if the Accumulator is reversed or not before the Final XOR.

Note: The Accumulator is not updated with the Final XOR value. If the comparison fails logic sets the $CRCERR$ flag and stops with $DONE$ flag also set.

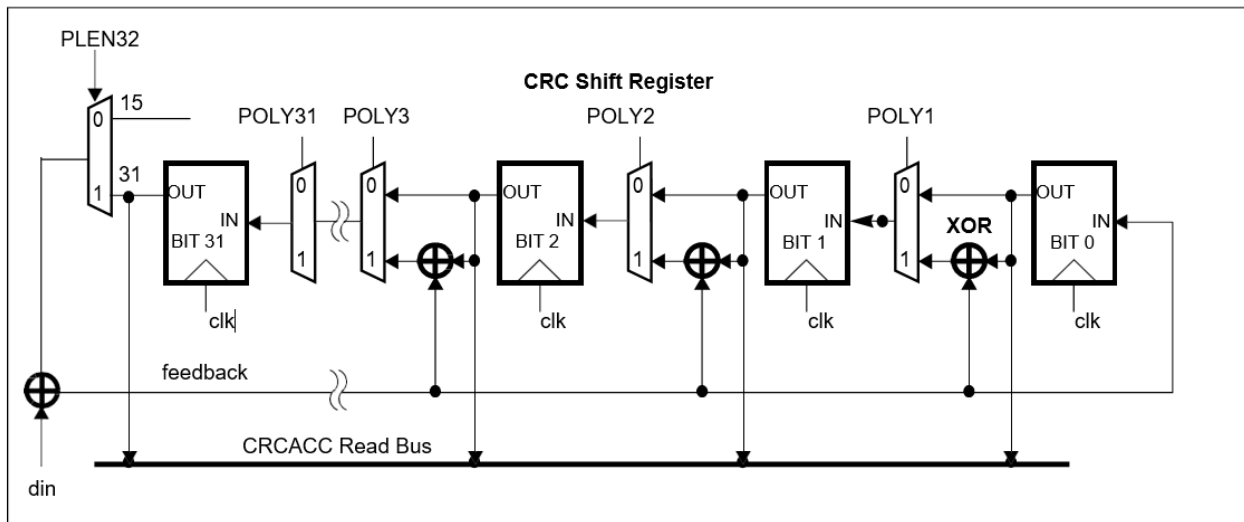
If no error occurs and $AUTOR$ is not set, the CRC logic sets $CRCDONE$ and $CRCERR$ remains cleared. If $AUTOR$ is set, then the CRC logic reloads all initial settings and restarts the CRC calculation without setting $CRCDONE$.

31.2.7.2 CRC Accumulator Topology

Other CRC modules have functions that support two selectable accumulator topologies. The topologies differ based on if $PLEN+1$ has trailing zeros in the byte stream or not. FCR's CRC topology does not have "extra" trailing zeros in its messages.

The LFSR Topology is specific to Flash-based communications and not streaming data. Thus this topology is used in this device. Its topology is shown in the following figure.

Figure 31-4. CRC LFSR Topology



31.2.7.3 Manual Usage Model

When the Auto Repeat bit selects manual mode, the last step of the CRC operation always sets the $CRCDONE$ bit. The state of $CRCERR$ reflects the validity of the message (flash memory region).

If checking the same region (i.e. message) again, simply clear $CRCDONE$ and $CRCERR$ before setting $CRCEN$ again. The IV loads into the ACC and the operation begins again.

31.2.7.4 Auto Repeat Usage Model

When the Auto Repeat bit selects repeat mode, the CRC operation only exits at the end of a loop if there is a comparison error. The error causes the logic to set both CRCDONE and CRCERR. To restart the CRC, simply clear CRCDONE and CRCERR before setting CRCEN again.

If there is no comparison error the logic restarts the CRC operation by loading the IV into ACC, the MLEN into the length counter, and Period into the Period counter. CRCDONE, CRCERR, and CRCEN remain unchanged. As mentioned above, each read waits the Period count including the first read of a loop. The CRC operation continues looping until it detects an error or is stopped manually.

31.2.7.5 CRC Interrupt Event

The CRCDONE bit reflects the state of the interrupt. When it is ONE the interrupt is asserted. Clearing CRCDONE de-asserts the interrupt. The system level interrupt controller determines if the CPU see the interrupt.

31.2.7.6 Performance

2.7.6 PERFORMANCE To process each byte of data (8-bits), the calculation takes one peripheral clock. However it reads a multi-byte flash word and operates on all those bytes each read. The Flash Word in Bytes, FWB, is either 16 or 32 bytes depending on the flash system. This yields either 16 or 32 clocks to calculate a

Since the CRC is reading the Flash it must also use the system Flash Wait States, FWS. The time to get the flash data in to the shift register is FWS+1. Also, if the system continues to access flash, the CRC logic is lowest priority and must wait until there is an idle cycle.

The logic periodically reads the flash to supply data to the calculation. The PERIOD is programmable from 0 to 4095 extra Period Clock (PerCLK) counts from the end of the calculation. The PerCLK is an 8MHz fixed frequency clock. Though selecting a period of zero clocks won't completely cut off the system flash bandwidth, it could be severely limiting. System bandwidth can be less than half under such conditions. Low period selections are useful during light loads or scheduled checks. When running with auto repeat, an appropriate period value is one that finishes the calculation before to the maximum user allotted time.

Lastly, the CRC needs to know the Message Length, MLEN, in bytes over which to perform the calculation. Note that neither the starting address of the message nor the message length need to be aligned to FWB.

31.2.7.6.1 CRC Performance Equation

First, calculate the following values:

$T_{M\text{CALC}}$ = CRC message calculation time

P_{APB} = $1/(\text{Peripheral Bus speed})$ = Peripheral Bus Period

P_{AHB} = $1/(\text{AHB Clock speed})$ = AHB Period

FWB = Flash Word in Bytes = 32

T_{STEP} = $FWB * P_{APB}$ = CRC step time

T_{PER} = $P_{PERCLK} * PERIOD + 2 * P_{BUS} * PERIOD$ = The Period between CRC flash accesses

For $PERIOD > 0$, $2 * P_{BUS} * PERIOD = 2 * P_{BUS}$

For $PERIOD = 0$, $2 * P_{BUS} * PERIOD = 0$

T_{ACC} = $(FWS+1) * P_{BUS}$ = Flash Access Time

P_{BUS} = Bus Clock Period (note: AHB or APB clock as they are the same)

P_{PERCLK} = Period of Period Clock = $1/8 \text{ MHz} = 125\text{ns}$

Second, solve the following for the Message Calculation Time:

$T_{M\text{CALC}} = (T_{STEP} + 2 * T_{ACC} + T_{PER}) * MLEN / FWB$

2*TACC includes the delay waiting for a prior read to finish, and depends on flash utilization. MLEN/FWB is the number of flash reads needed by the CRC engine. If data is not aligned to FWB it may take 1 more read.

If T_{MCALC} is known or has been calculated, we can solve for the PERIOD:

$$PERIOD = [(FWB/MLEN)*T_{MCALC} - (T_{STEP} + 2*T_{ACC} + 2*P_{BUS})]/P_{PERCLK}$$

(note: P_{BUS} = AHB or APB clock as they are the same)

31.2.7.6.2 CRC Performance Example

Assume the application must calculate the CRC (Period) on an 80Kbyte message every 800ms. Further, assume that both the AHB and APB are running at 100 MHz and that Flash needs four wait states.

First, establish the following:

$$FWB = 32\text{Bytes}$$

$$MLEN = 80\text{Kbytes} = 80\text{Kb} * 1024\text{Bytes/KB} = 81920\text{Bytes}$$

$$T_{MCALC} = 800\text{ms} = 800,000,000\text{ns}$$

$$P_{APB} = 1/100\text{MHz} = 10\text{ns}$$

$$P_{AHB} = 1/100\text{MHz} = 10\text{ns}$$

$$P_{PERCLK} = 125\text{ns}$$

$$FWS = 4$$

$$MLEN/FWB = 81920\text{Bytes}/32\text{Bytes} = 2560$$

$$FWB/MLEN = 1/2560$$

$$T_{STEP} = FWB * P_{APB} = 32 * 10\text{ns} = 320\text{ns}$$

$$T_{ACC} = (FWS + 1) * P_{BUS} = (4 + 1) * 10\text{ns} = 50\text{ns}$$

Second, solve for the value of PERIOD:

$$PERIOD = [(FWB/MLEN)*T_{MCALC} - (T_{STEP} + 2*T_{ACC} + 2*P_{BUS})]/P_{PERCLK}$$

$$= [(1/2560)*800,000,000\text{ns} - (320\text{ns} + 100\text{ns} + 20\text{ns})]/125\text{ns}$$

$$= ((1/2560)*8e8\text{ns} - 440\text{ns})/125\text{ns} = (312,500 - 440)/125 = 2496.48$$

$$= 2496 \text{ (rounding down)}$$

31.2.7.7 CRC Example Configurations

The following table shows an example CRC configuration with a polynomial size of 32. For each term (x^n) of the polynomial used, a corresponding bit in the CRCPOLY Register is set to '1'. This polynomial implements CRC-32-IEEE 802.3.

Table 31-5. Example 32-Bit Polynomial CRC Setup

Field Name	Bit Value	Description
CRCMODE.PLEN32	1	Polynomial Length is 32-bits
CRCPOLY	0000 0100 1100 0001 0001 1101 1011 011-	CRC polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

The following table shows an example CRC configuration with the polynomial size of 16. The value of CRCPOLY shows the 12th bit and the 5th bit set to '1', as required by the equation. The 0th bit required by the equation is always XOR'd.

Table 31-6. Example 16-Bit Polynomial CRC Setup

Field Name	Bit Value	Description
CRCMODE.PLEN32	0	Polynomial Length is 16-bits
CRCPOLY	0000 0000 0000 0000 0001 0000 0010 000-	CRC polynomial: $x^{16} + x^{12} + x^5 + 1$

31.2.7.8 Effects of ECC on CRC

CRC reads of the Flash obey the selected ECC mode of the device. If ECC is enabled the CRC uses the corrected data. If ECC is not enabled (either bypass or simple parity) the CRC uses the uncorrected data. ECC errors, either SEC or DED, are not reported to the system for CRC reads. CRC reads of the Flash also do not affect the FLT features or SFR bits.

31.2.7.9 Wait States Used by CRC

CRC always uses system wait states.

31.2.7.10 Effects of Programming on CRC

The CRC logic pauses automatically when a programming or erase operation prevents it from reading the Flash.

31.2.7.11 Effects of DBGCTRL on CRC

When DBGCTRL.CRCRUN = 1 the CRC Logic continues operation during Debug Mode. When DBGCTRL.CRCRUN = 0 the CRC Logic halts operation during Debug Mode.

31.2.7.12 Effects of Sleep Mode on CRC

When RUNSTDBY = 1 and STATUS.PRM=0 (Flash Auto Standby) the CRC Logic continues operation during Standby mode. Otherwise, CRC logic halts operations. When RUNSTDBY = 0 OR STATUS.PRM=1 (Flash Hibernate) the CRC Logic halts operation during Standby mode. Unlike PAUSE, the counters also stop.

Note: The PRM status can change on entry to Standby based on CTRLB.SLP[]. The CRC logic uses the state of PRM when the device is in Standby to determine if the CRC runs or not.

31.2.7.13 CRCCTRL – CRC CONTROL REGISTER

31.2.7.13.1 RUNSTDBY

RUNSTDBY determines if the CRC will be calculated when in Standby Mode. When RUNSTDBY = 0, the CRC is not calculated. CRC can only be calculated in Standby if STATUS.PRM = 0.

31.2.7.13.2 CRCEN

CRCEN is essentially the “enable” and “go” bit. All other CRC related SFR bits and fields must be setup prior to writing CRCEN to one. When CRCEN = 1 all other CRC bits are write protected except CRCEN, CRCRST, and PAUSE.CRCEN auto clears when the operation finishes (CRCDONE==1).

31.2.7.13.3 CRCRST

CRCRST resets the CRC SFR bits and returns the FSM to its idle state. The FSM clears CRCRST before returning to idle.

31.2.7.14 CRCCTRL SFR Description

31.2.7.14.1 PERIOD[11:0]

The PERIOD field defines the number of PerCLK counts that the CRC FSM waits from the last bit shifted in to the LFSR until the next read from flash. The request to count is synchronized between the PerCLK domain and the System clock domain. Therefore up to two PerCLK and two System clock periods are added to the period.

31.2.7.14.2 RIN

The RIN, Reflected Input, controls the shift direction of each byte. When RIN = 0, the FSM feeds the LSFR LSbit first. When RIN = 1, the MSbit is first into the LFSR. RIN does not swap bytes in a (32-bit) word or a Flash word, bytes flow from lowest addressed to highest addressed.

31.2.7.14.3 ROUT

The ROUT, Reflected Output, controls the bit order output of the CRC Accumulator. When ROUT = 0, the FSM XORS the CRCACC with CRCFXOR directly. When ROUT = 1, the FSM reverses the bit order of the CRCACC before it XORS it with CRCFXOR.

31.2.7.14.4 AUTOR

The AUTOR selects between Manual mode and Auto Repeat mode. Manual mode requires software to start the CRC engine for each check of the message allowing. Manual mode allows software to check several different messages. Auto Repeat mode does not require software to restart the CRC engine to check the message. However, this mode is limited to checking the same message repeatedly, only stopping on an error or by software disable of the CRC.

31.2.7.14.5 PLEN32

The PLEN32 bit determines the Polynomial Length, either CRC-16 (when zero) or CRC-32 (when one). PLEN32 also defines the effective length of the bits in CRCIV, CRCACC, CRCPOLY, CRCFXOR, and CRCSUM.

Other CRC functions may have a fully programmable PLEN from 0 to 31. These values define CRC lengths from 1 to 32. So PLEN32=0 and PLEN32=1 are equivalent to PLEN=15 and PLEN=31, respectively, on a fully programmable CRC.

31.2.7.15 CRCPAUSE SFR Description

The PAUSE bit halts CRC reads of Flash memory. The CRC FSM only checks the PAUSE bit when the PERIOD counter (being zero) triggers a read. The counter and the LFSR do not pause. When the FSM exits pause with the counter being zero, it requests a read of Flash to feed the LFSR.

Software set and clear of the PAUSE bit may occur without causing the FSM to actually pause operation. This type of interaction can occur during the time that the PERIOD counter is non-zero.

31.2.7.16 CRCMADR SFR Description

CRCMADR is the byte address of the first byte in the message to be checked. The CRC logic requires a physical system address to determine the region of Flash targeted: boot vs main, and Flash Panel 1 vs Flash Panel 2.

31.2.7.17 CRCMLLEN SFR Description

The CRCMLLEN field defines the message length in bytes. The Message Length is limited to the size of the largest contiguous region of Flash memory.

31.2.7.18 CRCIV SFR Description

The CRCIV defines the Initial Value which is loaded into the CRCACC at the beginning of the CRC calculation. Both Manual and Auto Repeat modes use the CRCMODE.PLEN32 control to determine the Initial Value size. When selecting CRC-16, bit 16 and above are not used.

31.2.7.19 CRCACC SFR Description

The CRCACC, CRC Accumulator, reports the LFSR state which changes during the calculation for each byte shifted in. Section [CRCSUM SFR Description](#) defines how the compare equation uses CRCACC when the CRC calculation completes. CRCMODE.PLEN32 controls the accumulator size. When selecting CRC-16, bit 16 and above are not used.

31.2.7.20 CRCPOLY SFR Description

The CRCPOLY holds the desired Polynomial for the CRC. The highest bit of the polynomial is not contained in CRCPOLY but is always set to one in the CRC logic. The lowest bit, CRCPOLY[0], is always one and not writable. PLEN32 controls the polynomial size. When selecting CRC-16, bit 16 and above are not used.

31.2.7.21 CRCFXOR SFR Description

CRCFXOR contains the value to XOR with CRCACC in the final calculation before comparing to CRCSUM. Section [CRCSUM SFR Description](#) defines how the compare equation uses CRCFXOR

when the CRC calculation completes. CRCMODE.PLEN32 controls the Final XOR size. When selecting CRC-16, bit 16 and above are not used.

31.2.7.22 CRCSUM SFR Description

CRCSUM contains the checksum value. CRCCTRL.PLEN32 controls the checksum size. When selecting CRC-16, bit 16 and above are not used.

The CRC Compare Equation is given by:

Compare is True if CRCSUM = 1.

$$\text{CRCSUM} = (\text{CRCACC} \text{ xor } \text{CRCFXOR})$$

Results

Recall as stated in the Overview, that if the comparison fails (CRCSUM = 0), logic sets the CRCERR flag and stops the process by setting the DONE flag.

If no error occurs (CRCSUM = 0):

If AUTOR is not set (calculations only perform once), the CRC logic sets CRCDONE and clears CRCERR. No further checking happens until CRCDONE is cleared and CRCEN is set to load the initial values.

If AUTOR is set (calculations repeat), then the CRC logic reloads all initial settings and restarts the CRC calculation for the next value without setting CRCDONE. Thus CRC checking will be continuously performed until either an error is encountered, or CRCEN is cleared.

31.2.7.23 CRC Bits in INTFLAG

31.2.7.23.1 CRCDONE

For manual CRC, the CRCDONE bit indicates that the operation is complete with all bits in the message acted upon. The CRCERR bit may, or may not, be set. For auto repeat CRC, the CRCDONE bit is only set when there is an error so CRCERR is also set. This condition also causes the looping to stop.

31.2.7.23.2 CRCERR

The CRCERR bit reports the status of the compare at the end of the CRC when all bits in the message finish shifting into the LFSR.

Recall that $\text{CRCSUM} = (\text{CRCACC} \text{ xor } \text{CRCFXOR})$. The CRC Error then becomes:

$$\text{CRCERR} = \neg \text{CRCSUM}$$

31.2.8 Error Correction Code (ECC) Support

31.2.8.1 Read While Write (RWW, also known as *Live Update*)

With two Flash panels in the system, it is possible to execute from one Flash panel while programming another (see [Concurrent Access](#)). Consequently, the ECC generation logic is duplicated to support ECC parity bit generation for data writes and ECC syndrome generation for data reads.

31.2.9 Error Correction Logic

There are two device configuration fields (i.e. fuses) stored in the Flash that are used for ECC control. The first is FECCCTL which is the reset value of the SFR based ECCCTRL.ECCCTL[1:0]. The second is FECCUNLCK which is the reset value of ECCCTRL.ECCUNLCK.

31.2.9.1 ECCCTRL SFR Description

31.2.9.1.1 ECCUNLCK

The ECC mode of the Flash can be locked for the duration of the program lifetime in Flash. When FECCUNLCK is 0, ECCUNLCK is also 0 and the selected ECC mode cannot be changed until Flash is updated. This option prevents undesired changing of the mode. When FECCUNLCK is 1 (the default erased state of the flash), ECCUNLCK and ECCCTL can be modified.

Note: If ECCUNLCK is 0, debug mode cannot override the ECC or error reporting via DBGCTRL.

31.2.9.1.2 ECCCTRL - 128/256-bit

The field ECCCTRL determines how the parity bits are used for Flash reads and writes. The four Modes - ECC Strict, Dynamic, ECC Write w/Dynamic Error and Bypass - affect reads and writes differently.

For the above modes, writes to the flash also write ECC Control Bits, CTL[2:0], which store whether ECC or Simple Parity was calculated on the data. Control Bits exists for each flash word (128-bit or 256-bit data). If the FCW performs a Single Write then the CTL is written with 3'b111 (i.e. not changed from the default erase value of the bits) for Parity. If the FCW performs a Quad Write, the CTL is written as 3'b000 for ECC. CTL[2:0] must be 3'b111 for Single Writes using Simple Parity since all CTL[] bits are not updated with a Single Write. CTL[2:0] is updated for Quad Writes so 3'b000 works for selecting ECC.

The [ECC Control Bits](#) table shows the allowed operations based on ECCCTRL and its effect on the CTL[2:0] field.

31.2.9.1.3 Strict ECC Writes with ECC Reads

If ECCCTRL[1:0] = 2'b00, ECC hardware is always active. This mode is also referred to as ECC On.

Only Quad Write programming is allowed in this mode, which always calculates and stores ECC values for the data. The FCW disables the Single Write Program command by making it a no-op if an attempt is made to execute it. Note Row Write uses multiple Quad Writes for programming.

Reads in this mode ignore Flash ECC Control bits CTL[2:0] and always perform ECC error checking and correction. This mode can generate ECC SEC events or ECC DED events.

Note: Reads of CFM uses Dynamic to support. The FCR forces ECC to be enabled for all calibration word reads during the reset sequence.

31.2.9.1.4 Dynamic Writes with Dynamic Reads

If ECCCTRL[1:0] = 2'b01, ECC hardware dynamically switches between ECC and Parity. This mode is also referred to as Dynamic ECC.

Both Quad Write and Single Write programming are allowed in this mode.

Reads in this mode obey Flash ECC Control bits CTL[2:0] to determine whether ECC or Parity is calculated on the data. This mode can generate ECC SEC events, and ECC DED interrupt, and Parity Errors that are reported via the ECC DED interrupt.

Note that Flash ECC Control bits CTL[2:0] uses a majority detect during reads to make it tolerant to single bit failures. A single bit failure of CTL[2:0] is reported via the ECC SEC interrupt, and only occurs in this mode. If an ECC SEC occurs at the same time as a CTL SEC, the error is reported on ECC SEC interrupt since both single bit failures get corrected. They also only count as one error with respect to SECCNT.

31.2.9.1.5 ECC Writes with Dynamic Reads - 128/256-bit

If ECCCTRL[1:0] = 2'b10, ECC hardware dynamically switches between ECC and Parity. However, only Quad Writes are allowed. Neither DED nor Parity Errors cause a Bus Error. The DERR bit is still set for uncorrectable errors. Otherwise, this mode behaves the same as in "Dynamic Writes with Dynamic Reads", previously.

31.2.9.1.6 Dynamic Writes with No Error Checks - 128/256-bit

If ECCCTRL[1:0] = 2'b11, ECC hardware is bypassed (for reads). This mode is also referred to as ECC Bypass.

Both Quad Write and Single Write programming are allowed in this mode.

Reads in this mode ignore Flash ECC Control bits CTL[2:0] and never check ECC or Parity. This mode never generates SEC, DED, or single word parity bus errors.

31.2.9.2 ECC Double Error Detected (FFLTSYN.DSERR)

When Read w/ ECC or Read w/Parity is active (i.e. not OFF/Bypass) and a read from flash memory results in an ECC DED, the FCR returns a bus error to the initiator. The error is returned in-band with the data and the initiator determines how to handle the corrupted data.

For the ECC DED, read data is loaded into the read buffer, prefetch buffer, or cache (depending on implementation) with a DED tag so subsequent reads of the address also cause bus errors. In the case of a prefetch, the error is only reported if the address is actually read.

Note: SERR and DERR are now combined into a single value (DSERR) tracking both errors. SERR now refers to the Flash SEC Interrupt Clear Enable value INTENCLR.SERR. However SERR and DERR may still be referenced in the datasheet as pertaining to Single or Double Errors.

31.2.9.3 Single Error Corrected (FFLTSYN.DSERR)

31.2.9.3.1

When ECC is active (i.e. not OFF/Bypass) and a read from Flash memory results in an ECC Single Error Corrected (SEC), the FCR reports it via an interrupt (when SECCNT=0) since it is not a critical error. Data in the read buffer, prefetch buffer, or cache (depending on implementation) is correct and no further ECC events are generated for reads that hit the buffer as long as that data is in the buffer. See Section 2.5 for SEC on access to VSS pages for a device family that has an HSM option.

Each read of the flash that results in an SEC causes the FCR to set FFLTSYN.DSERR when SECCNT[] = 0. If the count in SECCNT is non-zero the FCR decrements it and does not set DSERR. The FCR does not reload SECCNT[] when it is zero. Software must write the desired count each time it services the SEC interrupt.

When Dynamic ECC is active, an SEC can be caused by a bit error in CTL. If this is the case, a DED and an SEC can be caused by the same read.

Note: SERR and DERR are now combined into a single value (DSERR) tracking both errors. SERR now refers to the Flash SEC Interrupt Clear Enable value INTENCLR.SERR. However SERR and DERR may still be referenced in the datasheet as pertaining to Single or Double Errors.

31.2.9.4 Simple Parity Error - 128/256-bit

When Dynamic ECC is active and the read is from an address where CTL selects simple parity, a parity error causes the same behavior as an ECC DED. The error is reported in-band (if ECCCTL[1:0]=2'b01) and the FFLTSYN.DSERR bit is set.

Unlike a DED, a Bus Error exception is not returned to the initiator.

31.2.10 ECC Fault Control

The flash system contains a fault injection mechanism to allow periodic testing of the ECC logic and error event generation. It can be setup to test Parity, ECC SEC, and ECC DED data faults. For Dynamic mode, it can also inject faults in the ECC Control Bits. The registers ECCCTRL, FFLTCTRL, FFLTPTR, and FFLTADR provide control, while FFLTCAP, FFLTPAR and FFLTSYN provide status. Flash Fault registers are named FFLT* to distinguish them from other memory fault registers.

Note that if the system level ECCCTL bits are set for Bypass mode, the fault logic behaves as if it is in Dynamic Mode so that the functional safety code can be developed without generating system level errors.

The ECC Fault logic has two modes of operation: Injection and Capture. Injection mode allows user code to test the ECC logic by forcing faults into the Flash read or write paths. Fault injection behaves slightly differently depending on the ECCCTL mode. Capture mode allows user code to monitor read faults by capturing the first address at which an ECC or Parity error occurs.

31.2.10.1 Flash Panel ECC Organization

The following table shows the ECC Calculation Vector bit order with respect to flash data bits and parity bits. The first column is the ECC Calculation Vector. The next two columns map the Read/Write

Data and ECC Parity bits to the Calculation Vector. $V[n]$ defines the bit which is selected by $FFLT*PTR$, so $n=FFLT*PTR$.

Table 31-7. Flash ECC Vector

Vector Bits	Data Bits	ECC Parity Bits
$V[n]$	$D[n]$	$EP[n]$
0	-	0
1	-	1
2	-	2
3	0	-
4	-	3
7:5	3:1	-
8	-	4
15:9	10:4	-
16	-	5
31:17	25:11	-
32	-	6
63:33	56:26	-
64	-	7
71:65	63:57	-
127:72	119:63	-
128	-	8
136:129	127:120	-
255:137	246:128	-
256	-	9
265:257	255:247	-

The following table shows the ECCCTL and CTL field decode. The CTL field is stored in Flash and the fault logic can inject errors into it based on $FFLTCTRL.CTLFLT$.

Table 31-8. ECC Control Bits

Error Correction Mode	ECCCTL[1:0]	Write Value CTL[2:0]	Read Values CTL[2:0]	Operation
Bypass Mode	2'b11	3'b000	Don't Care	Quad Write w/ ECC Read w/o ECC Check
		3'b111	Don't Care	Single Write w/ Parity Read w/o Parity Check
ECC Write Dynamic Read Mode	2'b10	3'b000	3'b000 3'b001 3'b010 3'b100	Quad Write w/ ECC Read w/ ECC
		3'b111	3'b111 3'b110 3'b101 3'b011	Single Write Not Available Read w/ Parity

Table 31-8. ECC Control Bits (continued)

Error Correction Mode	ECCCTL[1:0]	Write Value CTL[2:0]	Read Values CTL[2:0]	Operation
Dynamic Mode	2'b01	3'b000	3'b000 3'b001 3'b010 3'b100	Quad Write w/ ECC Read w/ ECC
		3'b111	3'b111 3'b110 3'b101 3'b011	Single Write w/ Parity Read w/ Parity
ECC Strict Mode	2'b00	3'b000	Don't Care	Quad Write w/ ECC Read w/ ECC
		N/A	N/A	Single Write is not available All reads use ECC.

Note: If switching modes, Single Writes in Bypass or Dynamic cause SEC and DED errors in ECC mode. It is highly recommend to select and use only one Error Correction Mode.

31.2.10.2 Usage Model for Fault Injection in ECC Mode

Fault Injection occurs at only the Flash address selected by the SFR FFLTADR[31:0]. The FFLTCTRL field FLTMD[2:0] determines the type of fault injected, (single or double) and if it is for reads or writes of Flash. The two fields in the FFLTPTR register, FLT1PTR and FLT2PTR, point to the Vector bit or bits (as ordered in the Flash ECC Vector Table) to invert. Single fault injection always uses FLT1PTR.

Fault Injection always occurs between the ECC logic and the Flash. For writes this means that errors are injected after the ECC/Parity calculation but prior to the data write to Flash. For reads this means that errors are injected after the read from Flash but prior to the ECC/Parity calculation.

In FFLTPAR, the SFR fields SECIN and DEDIN capture the parity bits as they are read from the Flash. They have no meaning for writes as the bits are always driven to zeros (during reads and writes) for the calculation. The SFR fields SECOUT and DEDOUT capture the calculated value for either reads or writes.

In FFLTSYN, the SFR field SECSYN captures the syndrome of the read. This is the XOR of the SECIN with SECOUT. The SFR field DEDSYN captures the overall parity of the values read from Flash. If there is NO Overall Parity change this bit is ZERO. If there is an Overall Parity change this bit is ONE. The following table shows how each of the four conditions are determined. For SEC, the SECSYN points to the bit that was in ERROR.

Though the ECC/Parity mode bits CTL are not used in this mode, the CTLFLT bits are still effective. They will alter writes and reads of that field. However, since CTL is not used in ECC STRICT Mode, errors in CTL have no effect on the ECC calculation or correction.

In ECC Strict Mode, the fields PERR, CTLSTAT and CERR in FFLTSYN are meaningless.

Table 31-9. Error Decode

SECSYN	DEDSYN	SERR	DERR	Condition
Zero	0	0	0	No Error
Zero	1	1	0	DED parity Bit Error
Non-Zero	0	0	1	Double Error
Non-Zero	1	1	0	SEC – Data bit error corrected

31.2.10.3 Usage Model for Fault Injection in Dynamic Model

When ECCCTRL.ECCCTL selects ECC Strict or Dynamic mode, the FFLTCTRL.CTLFLT bits can also be used to inject a fault in the Dynamic ECC CTL bits. This operation is performed in parallel with data fault injection.

Though more than one CTL bit can be inverted, the logic is only capable of correcting single bit errors in the CTL field. For writes, the selected bit is stored in Flash inverted from the normal value. For reads the selected bit is inverted before it is used to determine the ECC mode.

The CTLSTAT field in FFLTSYN reports the captured values of CTL from the read or write that hit FFLTADR. The ECC Control Bits table shows the values of CTL for which the Flash system interprets ECC vs Parity reads and generates the values for writes. Note that injecting an error on the CTL value for a write does not change whether the write uses ECC or Simple Parity.

When performing a Quad Write operation the fault logic, control and status behave the same as in ECC Mode. When performing a read and FFLTSYN.CTLSTAT indicates “Read w/ ECC”, the fault logic, control and status also behave as in ECC Mode. However, FFLTSYN.CERR indicates if CTLSTAT has a bit failure.

When performing Single Write operations, the fault logic injects faults the same way as in ECC Mode. However, some parity bits are not used for Simple Parity (see the ECC Control Bits table) and do not affect the calculation when read. Also, the SEC* and DED* fields plus PERR, CLSTAT, CERR, DERR and SERR are “Don’t Care”.

When performing a read and FFLTSYN.CTLSTAT indicates “Read w/ Parity”, SEC*, DED*, DERR, SERR are “Don’t Care”. The value in FFLTSYN.PERR reflects the error status of each Write Word. If a PERR = 1, then a bus error is generated on the DED error path.

31.2.10.4 Usage Model for Fault Capture

When FFLTCTRL.FLTMD selects Fault Capture Mode, logic captures ECC/Parity status information. Capture only occurs when the INTFLAG.FLTCAP = 0, and never when it is ONE. In other words, once a fault capture has occurred any additional faults are ignored until INTFLAG.FLTCAP is cleared. Capture occurs for a read only on the detection of a fault which also cause logic to set INTFLAG.FLTCAP = 1. Capture occurs for each flash write but the INTFLAG.FLTCAP bit remains ZERO. This allows write ECC logic to be checked without interfering with fault events causing a read capture. Note that the write ECC/Parity logic is incapable of generating an error interrupt.

If there are simultaneous read errors, the error associated with AHB0 is captured. If AHB0 is not involved, then the error associated with panel 0 is captured and the error associated with panel 1 is dropped. If the simultaneous capture is between a read and a write, the read error is captured.

Only fields pertinent to the access type (ECC or Simple Parity) are valid. FFLTCTRL.FLTADR always captures the address at which the fault occurs (or for any write). CTLSTAT and CERR are valid for a DYNAMIC reads. If the access uses ECC then control logic updates SECSYN, DEDSYN, SERR, DERR, and all fields in FFLTPAR. If the access uses Simple Parity then control logic updates PERR.

31.2.10.5 System Effects on Fault Injection

Entry to Debug Mode or an Interrupt Service Routine can cause discrepancy between what the Fault Logic sees and what the CPU/system sees. This does not cause an erroneous condition for the system. However, the Fault Logic can show that an error occurred when it is not manifested in the system.

Because entry to these states are event driven and not part of the program flow the CPU can abort an instruction (or data) fetch in order to service the interruption. Fault code is susceptible to this condition if it happens to perform a read of Flash at this juncture.

31.2.10.5.1 Interrupts

When FLTMD selects one of the Fault Injection modes, fault logic does NOT set INTFLAG.FLTCAP. User code is expected to know it is accessing the flash at FFLTADR causing a fault to be injected. However, if the access is a read, user code can expect to experience a system error/interrupt

associated with the fault (.i.e. either an SEC interrupt or a bus error related to DED). Note that faults detected in DISABLED Mode do not cause system error/interrupts.

When FLTMD selects a Capture mode, fault logic sets INTFLAG.FLTCAP = 1 when it detects a CTL, SEC, or DED fault (captured and reported respectively by CERR, SERR, or DERR) from the flash read data path. Note that CTL faults detected in STRICT Mode (ECCCTL[1:0]=2'b00) do not set INTFLAG.FLTCAP

31.2.10.5.2 Effects on DBGCTRL on FLT

ECC and Fault capture are not controlled by DBGCTRL.CRCRUN. When the CPU is halted in debug mode the Flash ECC and Fault capture works as defined by DBGECC field in DBGCTRL.

31.2.10.6 FFLTCTRL SFR Description

31.2.10.6.1 FLTEN

When FLTEN = 0, the Fault Injection Logic is disabled and user code can safely update the SFR control bits.

When FLTEN = 1 the Fault Injection Logic performs the operation selected by FLTMD. User code can only modify FFLTCTRL when FLTEN = 1. A write any other Fault SFR results in a bus error.

The operation description of all subsequent bits are predicated on the Fault Logic being enabled.

31.2.10.6.2 CTLFLT

When FLTMD selects one of the Fault Injection modes, CTLFLT[2:0] controls fault injection on the Flash ECC Control bits CTL[2:0]. When CTLFLT[n] = 1 logic injects a fault in the associated bit position of CTL[n]. When CTLFLT[n] = 0 logic does not inject a fault on to the associated bit in CTL[n].

Control faults cannot be injected separately from parity faults. In all other FLTMD modes, CTLFLT[2:0] is unused. Also, CTLFLT is only meaningful for Dynamic ECC mode.

31.2.10.6.3 FLTMD

There are four basic fault modes: Disabled, Capture, Read Fault, and Write Fault. The selected mode takes affect when FLTEN = 1.

Fault Injection can be disabled, and write protected to ensure a fault is never injected in to the flash read or write path.

Fault Capture Mode allows the application to monitor the Flash for errors and take the appropriate action. Often times bit errors detected by ECC SEC are soft failures. They may be retention failures or read disturb failures which are correctable by re-writing the data. An application that wishes to decrease the likelihood of an uncorrectable ECC DED fault can reserve a spare page in Flash. The spare page provides a location to copy the offending page while correcting the SEC bit error(s). The offending page would then be erased and re-written with the original data intact. Fault Capture has a lock mode to prevent accidentally switching to Fault Injection Mode.

Fault Injection mode allows user code to periodically test the ECC logic to verify it is working correctly. There are four options. The first two inject a single or double fault for reads. The second two inject a single or double fault for writes. Depending on the fault testing requirements the write option may not be need for a single panel system since reads and writes share the same ECC calculation module. However, dual panel systems with Live Update never share the same ECC module.

31.2.10.7 FFLTPTTR SFR Description

31.2.10.7.1 FLT1PTR[8:0]

The Fault Pointer inverts the selected bit in the ECC Calculation Vector, V. For Single or Double Fault injection, logic inverts bit V[FLT1PTR], as defined in the Flash ECC Vector table. The table shows the mapping between the calculation vector and Data, ECC and Parity bits.

31.2.10.7.2 FLT2PTR[8:0]

The Fault Pointer inverts the selected bit in the ECC Calculation Vector. For Double Fault injection, logic inverts bit V[FLT2PTR], as defined in the Flash ECC Vector table. The table also shows the mapping between the calculation vector and Data, ECC and Parity bits.

31.2.10.8 FFLTADR SFR Description

FFLTADR.FLTADR contains the System Physical Flash Address at which a Fault Injection occurs.

31.2.10.9 FFLTCAP SFR Description

FFLTCAP.FLTADR contains the System Physical Flash Address at which a Fault Capture occurred.

31.2.10.10 FFLTPAR SFR Description

This register contains the values input to and output from the ECC Calculation Module. For reads the values are post fault injection but for writes they are pre-fault injection. This register does not contain valid information when using Simple Parity.

Note: In Capture Mode faults are not injected.

31.2.10.10.1 SECIN[8:0]

This is the SEC Parity for data read from the Flash panel. For a write this field is always ZERO. For a 128-bit data Flash SECIN[8] is unused.

31.2.10.10.2 DEDIN

This is the Overall Parity for data read from the Flash panel. For a write this field is always ZERO.

31.2.10.10.3 SECOUT[8:0]

This is the SEC Parity calculated on either the read or write data.

31.2.10.11 FFLT SYN SFR Description

This register contains Simple Parity and ECC Parity status. For reads the values are post fault injection but for writes they are pre-fault injection.

Note: In Capture Mode faults are not injected.

31.2.10.11.1 SECSYN[8:0]

SEC Syndrome represents the bitwise XOR of SECIN and SECOUT. SECSYN is not valid for any write or Simple Parity mode.

31.2.10.11.2 DEDSYN

DED Syndrome represents the overall parity from all data bits, SEC Parity bits and the DED Parity bit read from Flash. DEDSYN is not valid for any write or Simple Parity mode.

31.2.10.11.3 DERR and SERR

Double Error and Single Error status indicate the read state of the data: No Error, Single Error, or Double Error. The Error Decode table shows how DEDSYN and SECSYN determine their values. These bits are not valid for any write or Simple Parity mode.

31.2.10.11.4 CERR

Control Error indicates the single bit error status of the CTL[2:0] field. CERR is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

31.2.10.11.5 CTLSTAT[2:0]

CTLSTAT[2:0] reports the read value of the CTL [2:0] field stored in Flash for the addressed (FLTADR) Flash word (128-bit or 256-bit data). CTLSTAT is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

31.2.10.11.6 PERR[3:0]

Parity Error captures the difference between each word's read parity and its calculated parity. The word size is dependent on the Flash and is either 32-bit or 64-bit. PERR[0] represents the error

status of the lowest addressed word in of the Flash read, while PERR[3] is for the highest addressed word. PERR is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

31.2.11 Flash Read Protect Features

Flash read protection features are for the security of protected information. There are two types of read protection:

1. Reads to a page locked via CRP and cannot be read until the next reset. This typically refers to boot code including those done by the Boot ROM.
2. Device families that have an HSM option. Note that reads to a VSS page can only be accessed by the HSM.

Read protection controlled by the Flash system is directly protected within Flash system and does not require setup external to the FCR.

31.2.11.1 CFG Read Protect (CRP)

CRP *RP bits provide read protection for CFM pages. Some CFM pages may contain secret information accessible only to the BootROM code. The BootROM can protect and lock these page before it exits. The BootCfg page contains values to be used by the BootROM to program the CRP register before allowing boot flash to execute. See the device spec for this location do so. When read protected, read of this page cause a bus error.

If TrustZone is enabled on the device, then pages not used by the BootROM could be left unlock such that trusted code has the option to select the page settings. However, if the device is not a TrustZone device then the BootROM should lock the protection setting while keeping the page readable.

When read protected, reads of this space generate a bus error, return data of all 1's and do not report SEC/DED ECC or Parity errors. Internal sources such as CRC and FLT are blocked (via return data of all 1's) w/o generation of an access error.

31.2.11.1.1 Local Lock Bits

The user may optionally select to prevent further writes to a CRP register bits by also setting associated local LOCK bits (in the same register) when writing the CRP value.

When a local lock bit is set ($BC_nRPLOCK = 1$, $n = 1$ or 2), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a "write once" register. Local LOCK bits revert to the unlocked state at reset.

31.2.11.1.2 VSS Page Protect

For device families that have an HSM option, the VSS pages are always read protected from sources other than the HSM. The FCR protects VSS from internal source such as the CRC and FLT. CRC logic is blocked from reading VSS pages and FLT logic cannot inject or capture faults for reads or writes of VSS pages.

31.3 Peripheral Dependencies

Table 31-10. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
FCW	0x44004000	1 : FCW	AHB: CLKMSK[0] MASK2	---	2 STATUS0 PERID2			PD_CORE_SW

31.4 Peripheral Dependencies

Table 31-11. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
FCR	0x44002000	0 : FCR	AHB: CLKMSK[0] MASK1	---	1 STATUS0 PERID1			PD_CORE_SW

31.5 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0						PRIV		
0x04	CTRLP	31:24								
		23:16								
		15:8								
		7:0	PREPG				NVMOP[3:0]			
0x08	MUTEX	31:24								
		23:16								
		15:8								
		7:0						OWNER[1:0]		LOCK
0x0C	INTENCLR	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x10	INTENSET	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x14	INTFLAG	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x18	INTFLAGSET	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0x1C	STATUS	31:24								
		23:16								
		15:8								HTDRDY
		7:0								BUSY
0x20	KEY	31:24	CODE[23:16]							
		23:16	CODE[15:8]							
		15:8	CODE[7:0]							
		7:0	FEATURE[7:0]							
0x24	ADDR	31:24	ADDR[31:24]							
		23:16	ADDR[23:16]							
		15:8	ADDR[15:8]							
		7:0	ADDR[7:0]							
0x28	SRCADDR	31:24	SRCADDR[31:24]							
		23:16	SRCADDR[23:16]							
		15:8	SRCADDR[15:8]							
		7:0	SRCADDR[7:0]							
0x2C	DATA0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x30	DATA1	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x34	DATA2	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	DATA3	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x3C	DATA4	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x40	DATA5	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x44	DATA6	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x48	DATA7	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x4C	SWAP	31:24								
		23:16								
		15:8							PFSLOCK	PFSWAP
		7:0							BFSLOCK	BFSWAP
0x50	PWP0	31:24								PWPBASE[11:8]
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR				PWPSIZE[11:8]	
		7:0	PWPSIZE[7:0]							
0x54	PWP1	31:24								PWPBASE[11:8]
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR				PWPSIZE[11:8]	
		7:0	PWPSIZE[7:0]							
0x58	PWP2	31:24								PWPBASE[11:8]
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR				PWPSIZE[11:8]	
		7:0	PWPSIZE[7:0]							
0x5C	PWP3	31:24								PWPBASE[11:8]
		23:16	PWPBASE[7:0]							
		15:8	PWPEN	PWPLOCK	PWPMIR				PWPSIZE[11:8]	
		7:0	PWPSIZE[7:0]							
0x60 ... 0x6F	Reserved									
0x70	LBWP	31:24	LBWPLOCK							
		23:16								
		15:8	LBWP[15:8]							
		7:0	LBWP[7:0]							
0x74	UBWP	31:24	UBWPLOCK							
		23:16								
		15:8	UBWP[15:8]							
		7:0	UBWP[7:0]							
0x78	UOWP	31:24						UO2WPRLOCK[3:0]		
		23:16						UO1WPRLOCK[3:0]		
		15:8						UO2WPR[3:0]		
		7:0						UO1WPR[3:0]		
0x7C	CWP	31:24								
		23:16								
		15:8								
		7:0								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x80 ... 0x83	Reserved									
0x84	WPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPLCK	WPEN
0x88 ... 0x9F	Reserved									
0xA0	HSMINTENCLR	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0xA4	HSMINTENSET	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0xA8	HSMINTFLAG	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0xAC	HSMINTFLAGSET	31:24								
		23:16								
		15:8			WRERR	RSTERR				HTDPGM
		7:0	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
0xB0	HSMCWP	31:24	VSSWPLOCK5	VSSWPLOCK4	VSSWPLOCK3	VSSWPLOCK2				
		23:16			VSSWPLOCK1	VSSWPLOCK0				
		15:8	VSSWP5	VSSWP4	VSSWP3	VSSWP2				
		7:0			VSSWP1	VSSWP0				
0xB4	HSMWPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPLCK	WPEN

31.5.1 Control A REGISTER

Name: CTRLA
Offset: 0x0
Reset: 0x00000000
Property: R/W

Table 31-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						PRIV		
Access						R/W		
Reset						0		

Bit 2 – PRIV Privileged Access Only

Value	Name	Description
1	ENABLE	Macro registers only accessible in privileged accesses
0	DISABLE	Macro register accessible in privileged and unprivileged accesses.

31.5.2 Control Operation REGISTER

Name: CTRL0P
Offset: 0x4
Reset: 0x00000000
Property: R/W

Table 31-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PREPG				NVMOP[3:0]			
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0

Bit 7 – PREPG Note: This field can only be modified when BUSY=0 and KEY=<WRKEY Value> - same as NVMOP

Notes:

1. This field can only be modified when PAC Write protection is disabled, STATUS.BUSY=0, and KEY=<WRKEY Value> - same as NVMOP below.
2. It is recommended to either always use or never use PREPG.

Value	Name	Description
1	ENABLE	Program Operations include Pre-Program step
0	DISABLE	Program Operations exclude Pre-Program step

Bits 3:0 – NVMOP[3:0] Boot ROM Accessible Operations - Reserved Op in Mission mode Note: Ignores all write protect settings (CWP, BWP, PWP). Note: Chip Erase Operation is disabled by Boot ROM using BRCTRL.CEEN. Mission Mode Accessible Operations: 1111...1000 = RSVD For all Write/Erase Operations below, the entire target range must not be write protected. Condition: {PFM_DATA_MSB==63} 0001 = Reserved (don't use, implemented as DBLWRITE for testing purposes only) Note: See PWP*/*BWP for PFM/BFM write protection conditions.

1111-1000 = Reserved

For all Write/Erase Operations below, the entire target range must not be write protected.

0111 = PFM Erase Operation

0110 = Reserved

0101 = Reserved

0100 = Page Erase Operation: Erases page selected by ADDR

0011 = Row Write Operation: Programs row selected by ADDR

0010 = Quad (DWord) Program Operation: Programs flash word selected by ADDR

0001 = Single (DWord) Program Operation: Programs word selected by ADDR ⁽¹⁾

0000 = No Operation

Note:

1. If FCR.ECCCTRL.ECCCTL[1:0]=2'b00, this operation performs a No-Op but does not affect WRERR or RSTERR.

Value	Name	Description
14	CE	Chip Erase Operation: Erases PFM+SRF, BFM, CFM (except for CalOTP, Test, VSS pages if HSM is Present, UserOTP, & ROMCFG).
7	PFMERASE	PFM Erase Operation: Upper & Lower PFM Erase
6	UPFMERASE	Upper PFM Erase Operation: (RSVD NVMOP for Single Panel Devices)
5	LPFMERASE	Lower PFM Erase Operation: (RSVD NVMOP for Single Panel Devices)
4	PAGEERASE	Page Erase Operation: Erases page selected by ADDR
3	ROWWRITE	Row Write Operation: Programs row selected by ADDR
2	DBLWRITE	Double (Word) Program Operation: Programs double word selected by ADDR
0	NOP	No Operation
2	QUADWRITE	Quad (Word) Program Operation: Programs flash word selected by ADDR
1	SINGLEWRITE	Single (Word) Program Operation: Programs word selected by ADDR
0	NOP	No Operation
2	QUADWRITE	Quad (DWord) Program Operation: Programs flash word selected by ADDR
1	SINGLEWRITE	Single (DWord) Program Operation: Programs double word selected by ADDR
0	NOP	No Operation

31.5.3 MUTEX REGISTER

Name: MUTEX
Offset: 0x8
Reset: 0x00000000
Property: R/W

Note: If LOCK'd when a reset occurs, hardware reverts ownership to OWNER=00 and LOCK=0 after the controller cleans up the operation.

Table 31-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						OWNER[1:0]		LOCK
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – OWNER[1:0] Note: Only the HSM can write its owner ID. Only System (non-HSM) can write its owner ID.

Value	Name	Description
3	HSMFORCE	The HSM owns the FCW (by forcefully taking it from the System if necessary)
2	HSM	The HSM owns the FCW
1	SYSTEM	The System owns the FCW
0	NONE	No Owner - always 0 if LOCK=0
1	SYSTEM	The System owns the FCW
0	NONE	No Owner - always 0 if LOCK=0

Bit 0 – LOCK Note: This bit can only be cleared by the owner.

Value	Name	Description
1	LOCK	FCW is locked by owner
0	UNLOCK	FCW is not locked

31.5.4 Interrupt Enable Clear REGISTER

Name: INTENCLR
Offset: 0xc
Reset: 0x00000000
Property: R/K

Table 31-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			WRERR	RSTERR				HTDPGM
Access			R/K	R/K				R/K
Reset			0	0				0
Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Read Value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Write Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 12 – RSTERR Read Value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Reset or Brown Out Detect Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 8 – HTDPGM Read Value: Note: This field is unused if the system does not have an HTD.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 7 – SECERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Security Violation Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 6 – OPERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the NVMOP Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 5 – WPERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Write Protection Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 4 – BUSERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the AHB Bus Error During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 3 – FIFOERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the FIFO Underrun During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 2 – CFGERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Configuration Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled ($=1 \geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 1 – KEYERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Key Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled ($=1 \geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 0 – DONE Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the NVM Operation Done as an interrupt request.

Reading this bit returns whether this interrupt is enabled ($=1 \geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

31.5.5 Interrupt Enable Set REGISTER

Name: INTENSET
Offset: 0x10
Reset: 0x00000000
Property: R/S

Table 31-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			WRERR	RSTERR				HTDPGM
Access			R/S	R/S				R/S
Reset			0	0				0
Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Read Value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Write Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 12 – RSTERR Read Value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Reset or Brown Out Detect Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 8 – HTDPGM Read Value: Note: This field is unused if the system does not have an HTD.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 7 – SECERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Security Violation Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 6 – OPERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the NVMOP Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 5 – WPERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Write Protection Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 4 – BUSERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the AHB Bus Error During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 3 – FIFOERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the FIFO Underrun During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 2 – CFGERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Configuration Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled ($=1 \geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 1 – KEYERR Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Key Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled ($=1 \geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 0 – DONE Read Value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the NVM Operation Done as an interrupt request.

Reading this bit returns whether this interrupt is enabled ($=1 \geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

31.5.6 Interrupt Flag REGISTER

Name: INTFLAG
Offset: 0x14
Reset: 0x00000000
Property: R/K

Notes:

1. The interrupt flag bits of this register are set by hardware only.
2. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 31-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			WRERR	RSTERR				HTDPGM
Access			R/K/HS	R/K/HS				R/K/HS
Reset			0	0				0
Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		The Write/Erase sequence did not complete successfully
0		The Write/Erase sequence completed normally

Bit 12 – RSTERR The error is only captured during Write/Erase operations. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		A reset or Low Voltage Detected (possible data corruption, verify data)

Value	Name	Description
0		No Reset and Voltage level OK during write/erase

Bit 8 – HTDPGM This status is only captured for Write/Erase operations. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		High Temp Detected (possible data corruption, verify operation)
0		High Temp NOT Detected

Bit 7 – SECERR Attempted operation violates security configuration. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Security Violation Error
0		No Security Violation Error

Bit 6 – OPERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Selected Operation is Disabled Error
0		No NVMOP Error

Bit 5 – WPERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Write Protection Error
0		No Write Protection Error

Bit 4 – BUSERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Bus Error
0		No Bus Error

Bit 3 – FIFOERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		FIFO Error
0		No FIFO Error

Bit 2 – CFGERR Attempted Write/Erase when disallowed by a configuration setting. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		CFC Error
0		No CFG Error

Bit 1 – KEYERR Attempted to write to an SFR bit without first enabling it via KEY. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Key Error
0		No Key Error

Bit 0 – DONE When NVMOP completes the FSM clears BUSY and sets Done. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		NVMOP Done
0		NVMOP Not Done

31.5.7 Interrupt Flag Set REGISTER

Name: INTFLAGSET
Offset: 0x18
Reset: 0x00000000
Property: R/S

Notes:

1. The interrupt flag bits of this register are set by hardware only.
2. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 31-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			WRERR	RSTERR				HTDPGM
Access			R/S/HS	R/S/HS				R/S/HS
Reset			0	0				0
Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 12 – RSTERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 8 – HTDPGM Read value reflects state of INTFLAG. Write value:

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 7 – SECERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 6 – OPERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 5 – WPERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 4 – BUSERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 3 – FIFOERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 2 – CFGERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 1 – KEYERR Read value reflects state of INTFLAG. Write value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 0 – DONE Read value reflects state of INTFLAG. Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

31.5.8 Status REGISTER

Name: STATUS
Offset: 0x1c
Reset: 0x00000000
Property: R

Table 31-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
								HTDRDY
Access								R/HS/HC
Reset								0
Bit	7	6	5	4	3	2	1	0
								BUSY
Access								R/HS/HC
Reset								0

Bit 8 – HTDRDY Note: This field is always 1 if the system does not have an HTD.

Value	Name	Description
1	READY	HTD Ready
0	NOTREADY	HTD Not Ready

Bit 0 – BUSY NVM Busy Status

Note: This bit is read-only. It is set and cleared by hardware.

Value	Name	Description
1	BUSY	NVM Busy - All SFR bits are not writable
0	IDLE	NVM Not Busy

31.5.9 Feature Unlock Key REGISTER

Name: KEY
Offset: 0x20
Reset: 0x00000000
Property: R/W

Notes:

1. KEY can only be written when STATUS.BUSY=0.
2. KEY resets to 0 (a valid lock code) on a successful write to CTRLA, SWAP or an unsuccessful write to any register (any write that sets KEYERR).
3. KEY allows for multiple writes to configuration registers when using CFGKEY. To lock, write a Valid Lock Code to KEY.
4. Byte and Half-Word writes are ignored but result in a bus error.

Table 31-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CODE[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CODE[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CODE[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FEATURE[7:0]							
Access	R/W/HC	R/W/HC	R/W/HC	R/W/HC	R/W/HC	R/W/HC	R/W/HC	R/W/HC
Reset	0	0	0	0	0	0	0	0

Bits 31:8 – CODE[23:0] This register is used to lock and unlock write access to other Flash Contoller SFR bits. The KEY state remains active until it is written to a Valid Lock Code or an CTRL0P is written. 0x91C32C = Valid Unlock code - Unlocks selected SFR(s) Others = Valid Lock Code - Locks all SFR protected by KEY register
Note: KEY can only be written when BUSY=0. Note: KEY resets to 0 (a valid lock code) on a successful write to CTRL0P, SWAP or an unsuccessful write to any register (any write that sets KEYERR).

This register is used to lock and unlock write access to other FCW SFR bits.

The KEY state remains active until it is written to a Valid Lock Code or CTRL0P is written.

0x91C32C= Valid Unlock code - Unlocks selected SFR(s)

All Others = Valid Lock Code - Locks all SFR protected by KEY register

Bits 7:0 – FEATURE[7:0] This register is used to lock and unlock write access to other Flash Contoller SFR bits. Others = Invalid selection - Key Value set to 00 - Locks all SFR protected by KEY register
Note: KEY can only be

written when BUSY=0. Note: KEY reset to 0 (a valid lock code) on a successful write to CTRL0P, SWAP or an unsuccessful write to any register (any write that sets KEYERR).

This register is used to lock and unlock write access to other FCW SFR bits.

0x00= no selection, invalid selection, or invalid Key - Locks all SFR protected by KEY register

0x01= WRKEY Value: Unlock SFR bits associated with Write/Erase

0x02= SWAPKEY Value: Unlock SFR bits associated with Panel Swapping

0x04= CFGKEY Value: Unlock SFR bits associated with general Flash configuration

all other values: Invalid selection - Key Value set to 00 - Locks all SFR protected by KEY register

Value	Name	Description
0	LOCKALL	No selection, invalid selection, or invalid Key - Locks all SFR protected by KEY register
1	WRKEY	WRKEY Value: Unlock SFR bits associated with Write/Erase
2	SWAPKEY	SWAPKEY Value: Unlock SFR bits associated with Panel Swapping
4	CFGKEY	CFGKEY Value: Unlock SFR bits associated with general Flash configuration

31.5.10 Flash Destination Address REGISTER

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: R/W

Table 31-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – ADDR[31:0] This is a system byte address that the FCW aligns (by dropping lower ordered bits) to the minimum resolution of the NVMOP. Bulk/Chip/PFM Erase Address is ignored Page Erase Address identifies the page to erase Row Write Address identifies the row to write Condition: {PFM_DATA_MSB==63} Double Write (64-bits) Address identifies the Double Word to write. (ADDR[2:0] always read as 0.) Condition: {PFM_DATA_MSB==127} Single Write (32-bits) Address identifies the Word to write. (ADDR[1:0] always read as 0.) Quad Write (128-bits) Address identifies the QWord to write. (ADDR[3:0] are ignored, ADDR[1:0] always read 0.) Condition: {PFM_DATA_MSB==255} Single Write (64-bits) Address identifies the DWord to write. (ADDR[2:0] always read as 0.) Quad Write (256-bits) Address identifies the Quad DWord to write. (ADDR[4:3] are ignored, ADR[2:0] always read 0.) Note: This field can only be modified when BUSY=0.

31.5.11 System Source Address REGISTER

Name: SRCADDR
Offset: 0x28
Reset: 0x00000000
Property: R/W

Table 31-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SRCADDR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SRCADDR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SRCADDR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SRCADDR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SRCADDR[31:0] This is the system physical address of the data to be programmed into the flash when CTRL0P.NVMOP is set to Row Write. SRCADDR[1:0] are always 0. Note: This field can only be modified when BUSY=0.

This is the system physical address of the data to be programmed into the flash when CTRLA.NVMOP is set to Row Write SRCADDR[1:0] are always 0.

Note:

1. This field can only be modified when STATUS.BUSY=0.

31.5.12 Flash Write Data n REGISTER

Name: DATA
Offset: 0x2C + n*0x04 [n=0..7]
Reset: 0x00000000
Property: R/W

Table 31-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] The value in this register(s) is written to flash at the address defined in ADDR when a Write operation is commanded. Condition: {PFM_DATA_MSB==63} Double Write (64-bit data) Writes DATA0 to ADDR[31:3] with ADDR[2:0]=000 Writes DATA1 to ADDR[31:3] with ADDR[2:0]=100 Condition: {PFM_DATA_MSB==127} Single Write (32-bit data) Writes DATA0 to ADDR[31:2] with ADDR[1:0]=00 Quad Write (128-bit data) Writes DATA0 to ADDR[31:4], with ADDR[3:0]=0000 Writes DATA1 to ADDR[31:4], with ADDR[3:0]=0100 Writes DATA2 to ADDR[31:4], with ADDR[3:0]=1000 Writes DATA3 to ADDR[31:4], with ADDR[3:0]=1100 Condition: {PFM_DATA_MSB==255} Single Write (64-bit data) Writes DATA0 to ADDR[31:3], with ADDR[2:0]=000 Writes DATA1 to ADDR[31:3], with ADDR[2:0]=100 Quad Write (256-bit data) Writes DATA0 to ADDR[31:5], with ADDR[4:0]=0_0000 Writes DATA1 to ADDR[31:5], with ADDR[4:0]=0_0100 Writes DATA2 to ADDR[31:5], with ADDR[4:0]=0_1000 Writes DATA3 to ADDR[31:5], with ADDR[4:0]=0_1100 Writes DATA4 to ADDR[31:5], with ADDR[4:0]=1_0000 Writes DATA5 to ADDR[31:5], with ADDR[4:0]=1_0100 Writes DATA6 to ADDR[31:5], with ADDR[4:0]=1_1000 Writes DATA7 to ADDR[31:5], with ADDR[4:0]=1_1100 Note: This field can only be modified when BUSY=0.

The value in this register(s) is written to flash when a Write operation is commanded.

Single Write: (32-bit data)

Writes DATA0 to ADDR[31:2] with address bits [1:0] = 00

Quad Write: (128-bit data)

Writes DATA0 to ADDR[31:4], with address bits[3:0] = 0000

Writes DATA1 to ADDR[31:4], with address bits[3:0] = 0100

Writes DATA2 to ADDR[31:4], with address bits[3:0] = 1000

Writes DATA3 to ADDR[31:4], with address bits[3:0] = 1100

Note: This field can only be modified when STATUS.BUSY=0.

31.5.13 NVM Panel Swap REGISTER

Name: SWAP
Offset: 0x4c
Reset: 0x00000000
Property: R/W

Table 31-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							PFSLOCK	PFSWAP
Access							R/S	R/W
Reset							0	0
Bit	7	6	5	4	3	2	1	0
							BFSLOCK	BFSWAP
Access							R/S	R/W
Reset							0	0

Bit 9 – PFSLOCK Note: This field can only be modified when BUSY=0, PFSLOCK=0, and KEY=<SWAPKEY Value>
Writing a 0 to this bit has no effect.
Writing a 1 to this bit will set the lock bit.

Notes:

1. Once set, PFSLOCK can only be cleared by a reset.
2. This field can only be modified when STATUS.BUSY=0, PFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Name	Description
1	LOCKED	PFSLOCK and PFSWAP cannot be written
0	UNLOCKED	PFSLOCK and PFSWAP can be written

Bit 8 – PFSWAP Note: This field can only be modified when BUSY=0, PFSLOCK=0, and KEY=<SWAPKEY Value>
Note: This field can only be modified when STATUS.BUSY=0, PFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Name	Description
1	SWAP	Upper and Lower PFM Regions are Swapped
0	UNSWAP	Upper and Lower PFM Regions are NOT Swapped

Bit 1 – BFSLOCK Note: This field can only be modified when BUSY=0, BFSLOCK=0, and KEY=<SWAPKEY Value>

Writing a 0 to this bit has no effect.
Writing a 1 to this bit will set the lock bit.

Notes:

1. Once set, BFSLOCK can only be cleared by a reset.
2. This field can only be modified when STATUS.BUSY=0, BFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Name	Description
1	LOCKED	BFSLOCK and BFSWAP cannot be written
0	UNLOCKED	BFSLOCK and BFSWAP can be written

Bit 0 – BFSWAP Note: The BFSWAP value is determined by the values the user programmed SEQNUM in each boot panel.

Notes:

1. The BFSWAP value is determined by the values the user programmed SEQNUM in each boot panel.
2. This field can only be modified when STATUS.BUSY=0, BFSLOCK=0, and KEY.KEY=<SWAPKEY Value>.

Value	Name	Description
1	SWAP	Upper and Lower BFM Regions are Swapped
0	UNSWAP	Upper and Lower BFM Regions are NOT Swapped

31.5.14 PFM Write Protect Region n REGISTER

Name: PWP
Offset: 0x50 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: R/W

Table 31-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						PWPBASE[11:8]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PWPBASE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PWPEN	PWPLOCK	PWPMIR		PWPSIZE[11:8]			
Access	R/W	R/S	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PWPSIZE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:16 – PWPBASE[11:0] The Region Base Address is PWPBASE * 4096. When PWPEN=1, the region from PWPBASE*4096 to PWPBASE*4096+PWPSIZE+1 is write/erase protected. Note: This field can only be modified when PWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

The Region Base Address is PWPBASE * 4096.

When PWPEN=1, the region from PWPBASE to PWPBASE+PWPSIZE+1 is write/erase protected.

Note: This is a byte address force to align to page boundaries.

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Bit 15 – PWPEN Note: This field can only be modified when PWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Name	Description
1	ENABLE	PWP is Enabled for the defined region
0	DISABLE	PWP is Not Enabled for the defined region

Bit 14 – PWPLOCK Note: PWPLOCK can be set the same time as other bits are written.

Note: PWPLOCK can be set the same time as PWPBASE and PWPSIZE are written. Once set, PWPLOCK can only be cleared by a reset (Writing zero has no effect.).

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Name	Description
1	LOCKED	This register is Locked and cannot be modified
0	UNLOCKED	This register is Not Locked and can be modified

Bit 13 – PWPMIR Mirrors Lower PFM settings to Upper or Upper PFM settings to Lower. Used to maintain WP consistency between Upper and Lower PFM when using PFSWAP. Note: This field can only be modified when PWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>. Note: Mirroring is only valid for a dual panel flash system.

Mirrors Lower PFM settings to Upper or Upper PFM settings to Lower. This feature can be used to maintain Write Protect (WP) consistency between Upper and Lower PFM when using PFSWAP.

Note: When Mirrored, the PWPBASE address bit that distinguishes between Upper and Lower PFM is treated as a “Don’t Care”, meaning it can be 0 or 1.

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

Value	Name	Description
1	MIRROR	PWP settings are Mirrored
0	UNMIRROR	PWP settings are NOT Mirrored

Bits 11:0 – PWPSIZE[11:0] Region Size is (PWPSIZE+1) * 4096. 0x000 = 4KB 0x001 = 8KB ... 0xFFF = 16MB
Note: This field can only be modified when PWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

Pages Region Size is (PWPSIZE+1) *4KB.

0x000= 4KB

0x001= 8KB

...

0x3F= 256KB

Note: This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

31.5.15 Lower BFM Write Protect REGISTER

Name: LBWP
Offset: 0x70
Reset: 0x0000FFFF
Property: R/W

Table 31-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	LBWPLOCK							
Access	R/S							
Reset	0							
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	LBWP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	LBWP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – LBWPLOCK Note: This field can only be modified when LBWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

Notes:

1. This field can be set at the same time as writing to LBWP. Once set, LBWPLOCK can only be cleared by a reset (writing zero has no effect.).
2. This field can only be modified when LBWPLOCK = 0, STATUS.BUSY = 0 and KEY.KEY = <CFGKEY Value>.

Value	Name	Description
1	LOCKED	This register is Locked and cannot be modified
0	UNLOCKED	This register is NOT Locked and can be modified

Bits 15:0 – LBWP[15:0] Note: This field can only be modified when LBWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

Note:

1. This field can only be modified when LBWPLOCK = 0, STATUS.BUSY = 0 and KEY.KEY = <CFGKEY Value>.

Value	Name	Description
1	ENABLE	Erase and Write Protection for this Lower Boot Page is Enabled

Value	Name	Description
0	DISABLE	Erase and Write Protection for this Lower Boot Page is Disabled

31.5.16 Upper BFM Write Protect REGISTER

Name: UBWP
Offset: 0x74
Reset: 0x0000FFFF
Property: R/W

Table 31-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	UBWPLOCK							
Access	R/S							
Reset	0							
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	UBWP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	UBWP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit 31 – UBWPLOCK Note: This field can only be modified when UBWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

Notes:

1. This field can be set at the same time as writing to UBWP. Once set, UBWPLOCK can only be cleared by a reset (Writing zero has no effect.).
2. This field can only be modified when UBWPLOCK = 0, STATUS.BUSY = 0 and KEY.KEY = <CFGKEY Value>.

Value	Name	Description
1	LOCKED	This register is Locked and cannot be modified
0	UNLOCKED	This register is NOT Locked and can be modified

Bits 15:0 – UBWP[15:0] Note: This field can only be modified when UBWPLOCK=0, BUSY=0 and KEY=<CFGKEY Value>.

Note:

1. This field can only be modified when UBWPLOCK = 0, STATUS.BUSY = 0 and KEY.KEY = <CFGKEY Value>.

Value	Name	Description
1	ENABLE	Erase and Write Protection for this Upper Boot Page is Enabled

Value	Name	Description
0	DISABLE	Erase and Write Protection for this Upper Boot Page is Disabled

31.5.17 User OTP Write protection REGISTER

Name: UOWP
Offset: 0x78
Reset: 0x00000F0F
Property: R/W

Table 31-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						UO2WPRLOCK[3:0]		
Access					R/S	R/S	R/S	R/S
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
						UO1WPRLOCK[3:0]		
Access					R/S	R/S	R/S	R/S
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
						UO2WPR[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1
Bit	7	6	5	4	3	2	1	0
						UO1WPR[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					1	1	1	1

Bits 27:24 - UO2WPRLOCK[3:0] Note: This field can only be modified when UO2WPRLOCK[n]=0, BUSY=0 and KEY=<CFGKEY Value>.

Value	Name	Description
1	LOCKED	UO2WPR[n] & UO2WPRLOCK[n] bits are Locked and cannot be modified
0	UNLOCKED	UO2WPR[n] & UO2WPRLOCK[n] bits are NOT Locked and can be modified

Bits 19:16 - UO1WPRLOCK[3:0] Note: This field can only be modified when UO1WPRLOCK[n]=0, BUSY=0 and KEY=<CFGKEY Value>.

Writing a 0 to a bit has no effect.

Writing a 1 to a bit will enable the corresponding lock.

Notes:

1. This field can be set at the same time as writing to UO1WPR[n]. Once set, UO1WPRLOCK can only be cleared by a reset.
2. This field can only be modified when UO1WPRLOCK[n] = 0, STATUS.BUSY = 0, and KEY.KEY = <CFGKEY Value>.

Value	Name	Description
1	LOCKED	UO1WPR[n] & UO1WPRLOCK[n] bits are Locked and cannot be modified
0	UNLOCKED	UO1WPR[n] & UO1WPRLOCK[n] bits are NOT Locked and can be modified

Bits 11:8 – UO2WPR[3:0] Note: This field can only be modified when UO2WPRLOCK[n]=0, BUSY=0 and KEY=<CFGKEY Value>.

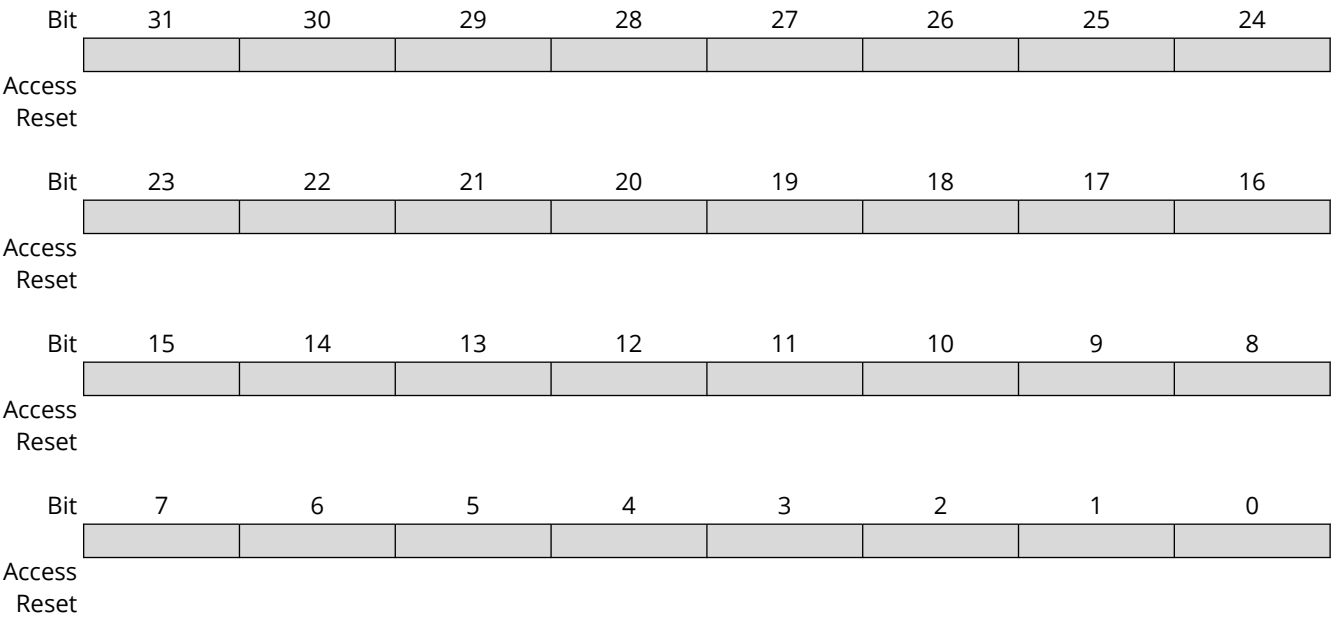
Value	Name	Description
1	ENABLE	Write Protection for User OTP Page 2 Row "n" is Enabled
0	DISABLE	Write Protection for User OTP Page 2 Row "n" is Disabled

Bits 3:0 – UO1WPR[3:0] Note: This field can only be modified when UO1WPRLOCK[n]=0, BUSY=0 and KEY=<CFGKEY Value>.

Value	Name	Description
1	ENABLE	Write Protection for User OTP Page 1 Row "n" is Enabled
0	DISABLE	Write Protection for User OTP Page 1 Row "n" is Disabled

31.5.18 CFM Page Write Protect REGISTER

Name: CWP
Offset: 0x7c
Reset: 0x000C000F
Property: R/W



31.5.19 Write Protection Control REGISTER

Name: WPCTRL
Offset: 0x84
Reset: 0x00000000
Property: R/W

Table 31-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPLCK	WPEN
Access							R/S	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Writing a value other than FCW_WPCTRL_KEY to this field cancels the write operation this register and generates a bus error. This field always returns 0 on read.

Bit 1 – WPLCK WPCTRL Write LOCK

Value	Name	Description
1	LOCKED	WPCTRL register is write protected. This bit can only be cleared by a hardware reset.
0	UNLOCKED	WPCTRL is not write protected.

Bit 0 – WPEN Note: If an HSM is present, HSM write protection is controlled by HSMWPCTRL.

Value	Name	Description
1	ENABLE	Register write protection enabled. Non-debugger writes to write-protected registers are canceled and generate a bus error. (Does not apply to any HSM register.)
0	DISABLE	Register write protection disabled.

31.5.20 HSM NVM Interrupt Enable Clear REGISTER

Name: HSMINTENCLR
Offset: 0xa0
Reset: 0x00000000
Property: R/K

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access			WRERR	RSTERR				HTDPGM
Reset			R/K	R/K				R/K
			0	0				0
Bit	7	6	5	4	3	2	1	0
Access	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Reset	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
	0	0	0	0	0	0	0	0

Bit 13 – WRERR Note: This register can only be modified when the initiator is HSM.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Write Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 12 – RSTERR Note: This field can only be modified when the initiator is HSM.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Reset or Brown Out Detect Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 8 – HTDPGM Note: This register can only be modified when the initiator is HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 7 – SECERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Security Violation Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 6 – OPERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the NVMOP Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 5 – WPERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Write Protection Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 4 – BUSERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the AHB Bus Error During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 3 – FIFOERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the FIFO Underrun During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 2 – CFGERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Configuration Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 1 – KEYERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Key Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 0 – DONE Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the NVM Operation Done as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY = 0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

31.5.21 HSM NVM Interrupt Enable Set REGISTER

Name: HSMINTENSET
Offset: 0xa4
Reset: 0x00000000
Property: R/S

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
			WRERR	RSTERR				HTDPGM
Access			R/S	R/S				R/S
Reset			0	0				0

Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/S	R/S	R/S	R/S	R/S	R/S	R/S	R/S
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Write Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 12 – RSTERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Reset or Brown Out Detect Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 8 – HTDPGM Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 7 – SECERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Security Violation Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 6 – OPERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the NVMOP Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 5 – WPERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Write Protection Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 4 – BUSERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the AHB Bus Error During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 3 – FIFOERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the FIFO Underrun During Row Write as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 2 – CFGERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Configuration Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 1 – KEYERR Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Key Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 0 – DONE Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the NVM Operation Done as an interrupt request.

Reading this bit returns whether this interrupt is enabled (=1 $\geq$ enabled).

Note: This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

31.5.22 HSM NVM Interrupt Flag REGISTER

Name: HSMINTFLAG
Offset: 0xa8
Reset: 0x00000000
Property: R/K

Note: The interrupt flag bits of this register are set by hardware only.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access			WRERR	RSTERR				HTDPMGM
Reset			R/K/HS	R/K/HS				R/K/HS
			0	0				0
Bit	7	6	5	4	3	2	1	0
Access	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Reset	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS	R/K/HS
	0	0	0	0	0	0	0	0

Bit 13 – WRERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		The Write/Erase sequence did not complete successfully
0		The Write/Erase sequence completed normally

Bit 12 – RSTERR The error is only captured during Write/Erase operations. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

The error is only captured during Write/Erase operations. Check the system RCAUSE register to see if this error was caused by a BOR event.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		A reset or Low Voltage Detected (possible data corruption, verify data)
0		No Reset and Voltage level OK during write/erase

Bit 8 – HTDPGM This status is only captured for Write/Erase operations. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		High Temp Detected (possible data corruption, verify operation)
0		High Temp NOT Detected

Bit 7 – SECERR Attempted operation violates security configuration. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Attempted operation violates security configuration.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		Security Violation Error
0		No Security Violation Error

Bit 6 – OPERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		Selected Operation is Disabled Error
0		No NVMOP Error

Bit 5 – WPERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted. Note: This register can only be modified when the initiator is HSM

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		Write Protection Error
0		No Write Protection Error

Bit 4 – BUSERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		Bus Error
0		No Bus Error

Bit 3 – FIFOERR Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		FIFO Error
0		No FIFO Error

Bit 2 – CFGERR Attempted Write/Erase when disallowed by a configuration setting. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Attempted Write/Erase when disallowed by a configuration setting.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		CFC Error
0		No CFG Error

Bit 1 – KEYERR Attempted to write to an SFR bit without first enabling it via KEY. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Attempted to write to an SFR bit without first enabling it via KEY.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		Key Error
0		No Key Error

Bit 0 – DONE When NVMOP completes the FSM clears BUSY and sets Done. Note: While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

When NVMOP completes the FSM clears BUSY and sets Done.

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will clear the flag.

Notes:

1. This field can only be modified when STATUS.BUSY=0, and the initiator is the HSM.
2. While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Value	Name	Description
1		NVMOP Done
0		NVMOP Not Done

31.5.23 HSM Interrupt Flag Set REGISTER

Name: HSMINTFLAGSET
Offset: 0xac
Reset: 0x00000000
Property: R/S

Table 31-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			WRERR	RSTERR				HTDPGM
Access			R/S/HS	R/S/HS				R/S/HS
Reset			0	0				0
Bit	7	6	5	4	3	2	1	0
	SECERR	OPERR	WPERR	BUSERR	FIFOERR	CFGERR	KEYERR	DONE
Access	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS	R/S/HS
Reset	0	0	0	0	0	0	0	0

Bit 13 – WRERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 12 – RSTERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 8 – HTDPGM Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 7 – SECERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 6 – OPERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 5 – WPERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 4 – BUSERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 3 – FIFOERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 2 – CFGERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 1 – KEYERR Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 0 – DONE Read value reflects state of HSMINTFLAG. Write value: Note: This register can only be modified when the initiator is HSM

Value	Name	Description
1		Set Interrupt Pending
0		No effect

31.5.24 HSM CFM Write Protect REGISTER

Name: HSMCWP
Offset: 0xb0
Reset: 0x0000F030
Property: R/W

Bit	31	30	29	28	27	26	25	24
	VSSWPLOCK5	VSSWPLOCK4	VSSWPLOCK3	VSSWPLOCK2				
Access	R/S	R/S	R/S	R/S				
Reset	0	0	0	0				

Bit	23	22	21	20	19	18	17	16
			VSSWPLOCK1	VSSWPLOCK0				
Access			R/S	R/S				
Reset			0	0				

Bit	15	14	13	12	11	10	9	8
	VSSWP5	VSSWP4	VSSWP3	VSSWP2				
Access	R/W	R/W	R/W	R/W				
Reset	1	1	1	1				

Bit	7	6	5	4	3	2	1	0
			VSSWP1	VSSWP0				
Access			R/W	R/W				
Reset			1	1				

Bits 28, 29, 30, 31 – VSSWPLOCK Note: This field can only be modified when This Bit(n)=0, BUSY=0, KEY=<CFGKEY Value>, and the initiator is HSM.

Value	Name	Description
1	LOCKED	The Lock & Write Protect bits for this Page are Locked and cannot be modified
0	UNLOCKED	The Lock & Write Protect bits for this Page are NOT Locked and can be modified
1	LOCKED	The Lock & Write Protect bits for this Page are Locked and cannot be modified
0	UNLOCKED	The Lock & Write Protect bits for this Page are NOT Locked and can be modified

Bits 20, 21 – VSSWPLOCK Note: This field can only be modified when This Bit(n)=0, BUSY=0, KEY=<CFGKEY Value>, and the initiator is HSM.

Value	Name	Description
1	LOCKED	The Lock & Write Protect bits for this Page are Locked and cannot be modified
0	UNLOCKED	The Lock & Write Protect bits for this Page are NOT Locked and can be modified
1	LOCKED	The Lock & Write Protect bits for this Page are Locked and cannot be modified
0	UNLOCKED	The Lock & Write Protect bits for this Page are NOT Locked and can be modified

Bits 12, 13, 14, 15 – VSSWP Note: This field can only be modified when its Lock Bit=0, BUSY=0, KEY=<CFGKEY Value>, and the initiator is HSM.

Value	Name	Description
1	ENABLE	Erase and Write Protection for this Page is Enabled
0	DISABLE	Erase and Write Protection for this Page is Disabled
1	ENABLE	Erase and Write Protection for this Page is Enabled
0	DISABLE	Erase and Write Protection for this Page is Disabled

Bits 4, 5 – VSSWP Note: This field can only be modified when its Lock Bit=0, BUSY=0, KEY=<CFGKEY Value>, and the initiator is HSM.

Value	Name	Description
1	ENABLE	Erase and Write Protection for this Page is Enabled
0	DISABLE	Erase and Write Protection for this Page is Disabled
1	ENABLE	Erase and Write Protection for this Page is Enabled
0	DISABLE	Erase and Write Protection for this Page is Disabled

31.5.25 HSM Write Protection Control REGISTER

Name: HSMWPCTRL
Offset: 0xb4
Reset: 0x00000000
Property: R/W

Table 31-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPLCK	WPEN
Access							R/S	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Writing a value other than FCW_WPCTRL_KEY to this field cancels the write operation this register and generates a bus error. This field always returns 0 on read.

Bit 1 – WPLCK WPCTRL Write LOCK

Value	Name	Description
1	LOCKED	HSMWPCTRL register is write protected. This bit can only be cleared by a hardware reset.
0	UNLOCKED	HSMWPCTRL is not write protected.

Bit 0 – WPEN Write Protection Enable

Value	Name	Description
1	ENABLE	Register write protection enabled for HSM accesses for all registers (not just those prefixed with HSM) with the Write-Protected property.
0	DISABLE	Register write protection disabled for HSM accesses.

31.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								ARB
		23:16					RDBUFWS[3:0]			
		15:8	AUTOWS	ADRWS			FWS[3:0]			
		7:0						PRIV		
0x04	CTRLB	31:24								
		23:16							PREFDIS	CHEDIS
		15:8							SLP[1:0]	
		7:0	CHEINV						TEMP	PRM
0x08	INTENCLR	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0								SERR
0x0C	INTENSET	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0								SERR
0x10	INTFLAG	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0								SERR
0x14	INTFLAGSET	31:24								
		23:16								FLTCAP
		15:8							CRCERR	CRCDONE
		7:0								SERR
0x18	STATUS	31:24								
		23:16								
		15:8								
		7:0							TEMP	PRM
0x1C	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0						DBGEC[1:0]		CRCRUN
0x20	ECCCTRL	31:24								
		23:16								
		15:8	SECCNT[7:0]							
		7:0	ECCUNLCK		ECCCTL[1:0]					
0x24	CRCCTRL	31:24								
		23:16								
		15:8								
		7:0			RUNSTDBY				CRCEN	CRCRST
0x28	CRCMODE	31:24					PERIOD[11:8]			
		23:16	PERIOD[7:0]							
		15:8	RIN	ROUT	AUTOR	PLEN32				
		7:0								
0x2C	CRCPAUSE	31:24								
		23:16								
		15:8								
		7:0								PAUSE
0x30	CRCMADR	31:24	CRCMADR[27:24]							
		23:16	CRCMADR[23:16]							
		15:8	CRCMADR[15:8]							
		7:0	CRCMADR[7:0]							
0x34	CRCMLEN	31:24								
		23:16	CRCMLEN[23:16]							
		15:8	CRCMLEN[15:8]							
		7:0	CRCMLEN[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	CRCIV	31:24	CRCIV[31:24]							
		23:16	CRCIV[23:16]							
		15:8	CRCIV[15:8]							
		7:0	CRCIV[7:0]							
0x3C	CRCACC	31:24	CRCACC[31:24]							
		23:16	CRCACC[23:16]							
		15:8	CRCACC[15:8]							
		7:0	CRCACC[7:0]							
0x40	CRCPOLY	31:24	CRCPOLY[31:24]							
		23:16	CRCPOLY[23:16]							
		15:8	CRCPOLY[15:8]							
		7:0	CRCPOLY[7:0]							
0x44	CRCFXOR	31:24	CRCFXOR[31:24]							
		23:16	CRCFXOR[23:16]							
		15:8	CRCFXOR[15:8]							
		7:0	CRCFXOR[7:0]							
0x48	CRCSUM	31:24	CRCSUM[31:24]							
		23:16	CRCSUM[23:16]							
		15:8	CRCSUM[15:8]							
		7:0	CRCSUM[7:0]							
0x4C	FFLTCTRL	31:24								
		23:16								
		15:8								
		7:0							FLTEN	FLTRST
0x50	FFLTMODE	31:24								
		23:16								
		15:8		FLTMD[2:0]				CTLFLT[2:0]		
		7:0								
0x54	FFLTPTR	31:24								FLTPTR2[8]
		23:16	FLTPTR2[7:0]							
		15:8								FLTPTR1[8]
		7:0	FLTPTR1[7:0]							
0x58	FFLTADR	31:24					FLTADR[27:24]			
		23:16	FLTADR[23:16]							
		15:8	FLTADR[15:8]							
		7:0	FLTADR[7:0]							
0x5C	FFLTCAP	31:24					FLTADR[27:24]			
		23:16	FLTADR[23:16]							
		15:8	FLTADR[15:8]							
		7:0	FLTADR[7:0]							
0x60	FFLTPAR	31:24	DEDOUT							SECOUT[8]
		23:16	SECOUT[7:0]							
		15:8	DEDIN							SECIN[8]
		7:0	SECIN[7:0]							
0x64	FFLTSYN	31:24	PERR[3:0]					CTLSTAT[2:0]		
		23:16						CERR	DSERR[1:0]	
		15:8	DEDSYN							SECSYN[8]
		7:0	SECSYN[7:0]							
0x68	CRP	31:24								
		23:16								
		15:8								
		7:0								
0x6C ... 0x6F	Reserved									
0x70	WPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0					WPFFLTEN	WPCRCEN	WPLCK	WPEN

31.6.1 CTRL A REGISTER

Name: CTRLA
Offset: 0x0
Reset: 0x00008000
Property: R/W

Table 31-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								ARB
Access								R/W
Reset								0
Bit	23	22	21	20	19	18	17	16
					RDBUFWS[3:0]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
	AUTOWS	ADRWS			FWS[3:0]			
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	1	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
						PRIV		
Access						R/W		
Reset						0		

Bit 24 – ARB AHB Arbitration scheme

Value	Name	Description
1	FIXED	Fixed priority AHB0 highest to AHB<FCR_AHB_PORTS_NB-1> lowest
0	ROUNDROBIN	Round Robin Arbitration

Bits 19:16 – RDBUFWS[3:0] Read Buffer deterministic access requires AUTOWS=0.

When returning data from the AHB read buffer, insert wait states to match ADRWS and FWS from Flash based on RDBUFWS.

Value	Name	Description
1	DETERMINISTIC	ADRWS + FWS wait state for hits to AHB read buffer
0	ZEROWS	0 Wait States

Bit 15 – AUTOWS Taws = Tacc + 5ns + 2 clocks

Taws = Tacc + 5ns + 2 clocks

Value	Name	Description
1	ENABLE	Use Automatic wait states: Total flash wait states are ADRWS + Taws
0	DISABLE	Use FWS: Total flash wait states are ADRWS + FWS

Bit 14 – ADRWS For Total flash wait states see AUTOWS.
For Total flash wait states see AUTOWS.

Value	Name	Description
1	ONE	Add 1 Address Wait State - allowing for higher clock frequencies
0	ZERO	Add 0 Address Wait States - allowing for higher performance at lower clock frequencies

Bits 11:8 – FWS[3:0] For Total flash wait states see AUTOWS. ... Note: This is not the wait states seen by the CPU.

1111= Fifteen Wait States
1110= Fourteen Wait States

...

0001= One Wait State
0000= Zero Wait States

Note: This is not the wait states seen by the CPU. For Total Flash wait states see AUTOWS.

Value	Name	Description
15		Fifteen Wait States
14		Fourteen Wait States
1		One Wait State
0		Zero Wait States

Bit 2 – PRIV Privileged Access Only

Value	Name	Description
1	ENABLE	Macro registers only accessible in privileged accesses
0	DISABLE	Macro register accessible in privileged and unprivileged accesses.

31.6.2 CTRL B REGISTER

Name: CTRLB
Offset: 0x4
Reset: 0x00000000
Property: R/W

Table 31-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
							PREFDIS	CHEDIS
Access							R/W	R/W
Reset							0	0
Bit	15	14	13	12	11	10	9	8
							SLP[1:0]	
Access							R/W	R/W
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	CHEINV						TEMP	PRM
Access	R/S/HC						R/S/HC	R/S/HC
Reset	0						0	0

Bit 17 – PREFDIS When CTRLB.CHEDIS==0: When CTRLB.CHEDIS==1: Predictive Prefetch is always disabled and this field is ignored.

When CTRLB.CHEDIS=0:

1 (DISABLE) = Predictive Prefetch Disabled

0 (ENABLE) = Predictive Prefetch Enabled

When CTRLB.CHEDIS=1:

Predictive Prefetch is always disabled and this field is ignored.

Value	Name	Description
1	DISABLE	Predictive Prefetch Disabled
0	ENABLE	Predictive Prefetch Enabled

Bit 16 – CHEDIS Cache Disable

Value	Name	Description
1	DISABLE	Cache Disabled; Interface uses RDBUFWS selection.
0	ENABLE	Cache Enabled; Accesses to flash use AUTOWS/FSW selection.

Bits 9:8 – SLP[1:0] Note: The status of power reduction is reflected in STATUS.PRM.

Note: The status of power reduction is reflected in STATUS.PRM.

Value	Name	Description
3	AUTOSLOW	Enter Flash Hibernate on entry to Standby w/ wakeup on first access
2	AUTOFAST	Enter Flash Hibernate on entry to Standby w/ wakeup to Auto Standby
1	RSVD01	RSVD
0	MANUAL	Maintain state of STATUS.PRM on entry to Standby

Bit 7 – CHEINV Cache InvalidateCache invalidation takes 1 clock cycle, but may be delayed by an active read. Cache invalidation takes 1 clock cycle, but may be delayed by an active read.

Value	Name	Description
1	INV	Invalidate Cache
0	NONE	No Action

Bit 1 – TEMP When written to 1, this bit remains a 1 until STATUS.TEMP toggles. Then it returns to 0. Note: When changing TEMP, software must adjust CTRLA.FWS appropriately or have CTRLA.AUTOWS==1. Note: The status of Operating Temp Read Mode is reflected in STATUS.TEMP.

Value	Name	Description
1	TOGGLE	Toggle the Operating Temperature Read Mode
0	NONE	No Action

Bit 0 – PRM When written to 1, this bit remains a 1 until STATUS.PRM toggles. Then it returns to 0. Note: The status of power reduction is reflected in STATUS.PRM.

When written to 1, this bit remains a 1 until STATUS.PRM toggles. Then it is cleared by hardware. Writing a 0 has no effect.

Note: The status of power reduction is reflected in STATUS.PRM.

Value	Name	Description
1	TOGGLE	Toggle the Flash Power Mode
0	NONE	No Action

31.6.3 Interrupt Enable Clear REGISTER

Name: INTENCLR
Offset: 0x8
Reset: 0x00000000
Property: R/K

Table 31-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/K
Reset								0
Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/K	R/K
Reset							0	0
Bit	7	6	5	4	3	2	1	0
								SERR
Access								R/K
Reset								0

Bit 16 – FLTCAP Read value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the ECC Fault Capture as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 9 – CRCERR Read value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the CRC Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 8 – CRCDONE Read value:

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the CRC Calculation Done as an interrupt request.

Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 0 – SERR Read value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will disable the Flash SEC as an interrupt request.

Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

31.6.4 Interrupt Enable SET REGISTER

Name: INTENSET
Offset: 0xc
Reset: 0x00000000
Property: R/S

Table 31-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/S
Reset								0
Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/S	R/S
Reset							0	0
Bit	7	6	5	4	3	2	1	0
								SERR
Access								R/S
Reset								0

Bit 16 – FLTCAP Read value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the ECC Fault Capture as an interrupt request.

Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 9 – CRCERR Read value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the CRC Error as an interrupt request.

Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 8 – CRCDONE Read value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the CRC Calculation Done as an interrupt request.

Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

Bit 0 – SERR Read value:

Writing a '0' to this bit has no effect.

Writing a 1 to this bit will enable the Flash SEC as an interrupt request.

Reading this bit returns whether this interrupt is enabled.

Value	Name	Description
1		Interrupt Enabled
0		Interrupt Disabled

31.6.5 Interrupt Flag REGISTER

Name: INTFLAG
Offset: 0x10
Reset: 0x00000000
Property: R/K

Table 31-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/K/HS
Reset								0
Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/K/HS	R/K/HS
Reset							0	0
Bit	7	6	5	4	3	2	1	0
								SERR
Access								R/K/HS
Reset								0

Bit 16 – FLTCAP Read value:

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

Value	Name	Description
1		An ECC Fault Capture, related to FLTMD[], occurred
0		No Interrupt Pending

Bit 9 – CRCERR Read value:

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

Value	Name	Description
1		CRCACC Is Not Equal to the XOR of CRCSUM and CRCFXOR
0		No Interrupt Pending

Bit 8 – CRCDONE Read value:

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the interrupt pending.

Value	Name	Description
1		Calculation Done
0		No Interrupt Pending

Bit 0 – SERR Read value:

Reading this bit returns the INTFLAG bit value.

Writing a '0' to this disables the pending interrupt.

Writing a '1' to this bit will set the interrupt pending, as if SECCNT count is reached.

Value	Name	Description
1		SECCNT Count reached
0		No Interrupt Pending

31.6.6 Interrupt Flag Set REGISTER

Name: INTFLAGSET
Offset: 0x14
Reset: 0x00000000
Property: R/S

Table 31-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								FLTCAP
Access								R/S/HS
Reset								0
Bit	15	14	13	12	11	10	9	8
							CRCERR	CRCDONE
Access							R/S/HS	R/S/HS
Reset							0	0
Bit	7	6	5	4	3	2	1	0
								SERR
Access								R/S/HS
Reset								0

Bit 16 – FLTCAP Read value reflects state of INTFLAG. Write value:
Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 9 – CRCERR Read value reflects state of INTFLAG. Write value:
Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 8 – CRCDONE Read value reflects state of INTFLAG. Write value:
Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the interrupt pending.

Value	Name	Description
1		Set Interrupt Pending
0		No effect

Bit 0 – SERR Read value reflects state of INTFLAG. Write value:

Reading this bit returns the INTFLAG bit value.

Writing a '0' to this disables the pending interrupt.

Writing a '1' to this bit will set the interrupt pending, as if SECCNT count is reached.

Value	Name	Description
1		SECCNT Count reached
0		No Interrupt Pending

31.6.7 Status REGISTER

Name: STATUS
Offset: 0x18
Reset: 0xE0000000
Property: R

Table 31-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							TEMP	PRM
Access							R/HS/HC	R/HS/HC
Reset							0	0

Bit 1 – TEMP Note: BootROM may update this field at reset based on user configuration.

Value	Name	Description
1		Flash is configured for High Temperature, High Latency reads
0		Flash is configured for Standard Temp, Low Latency reads

Bit 0 – PRM NVM Power Reduction Mode Status

Value	Name	Description
1		Flash is Powered Down and wakes on first access (read or NVMOP)
0		Flash is Ready (for read or NVMOP)

31.6.8 Debug Control CTRL REGISTER

Name: DBGCTRL
Offset: 0x1c
Reset: 0x00000000
Property: R/W

Table 31-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						DBGECC[1:0]		CRCRUN
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 2:1 – DBGECC[1:0] Debug ECC ModeECC errors from debugger reads are:
ECC errors from debugger reads are:

Value	Name	Description
3	RSVD11	Don't use but same as 01 (DISABLE)
2	ENABLE	Corrected per ECCCTRL; Bus Error, INTFLAG, SEC counter, and FLT logic operates as setup.
1	DISABLE	Not corrected; No Bus Error; INTFLAG, SEC counter, and FLT logic are not updated.
0	NOERR	Corrected per ECCCTRL; No Bus ERR; INTFLAG, SEC counter, and FLT logic are not updated.

Bit 0 – CRCRUN CRC Debug Run Enable

Value	Name	Description
1	RUN	CRC Logic Runs in Debug Mode.
0	HALT	CRC Logic Halts in Debug Mode

31.6.9 ECC Control REGISTER

Name: ECCCTRL
Offset: 0x20
Reset: 0x00000040
Property: R/W

Table 31-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access	SECCNT[7:0]							
Reset	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access		ECCUNLCK	ECCCTL[1:0]					
Reset		R/K	R/W	R/W				
Reset		1	0	0				

Bits 15:8 – SECCNT[7:0] SECCNT is the start value of an internal counter that decrements (by 1 per panel reporting an SEC) its count value each time an SEC occurs on a flash access (including the CTL[2:0] bits if in Dynamic ECC Mode). The internal counter stops decrementing at zero. If an SEC error occurs when the internal counter is zero, the SERR flag bit is set. Note: This field counts all SEC errors and is not limited to SEC errors on unique addresses.

SECCNT is the start value of an internal counter that decrements by 1 for each occurrence of a SEC event (including ECC CTL[2:0] bit if in Dynamic ECC Mode). The SECCNT counter stops decrementing at zero. If an SEC error occurs when the SECCNT counter is zero, the INTFLAG.SERR flag bit is set.

Note: This field counts all SEC errors and is not limited to SEC errors on unique addresses.

Bit 6 – ECCUNLCK The read value dictates the unlock state. Note: This field can only be modified when ECCUNLCK=1

Note: This field can only be modified when ECCUNLCK=1.

The read value dictates the unlock state:

Value	Name	Description
1	UNLOCKED	ECCUNLCK and ECCCTL[1:0] can be written
0	LOCKED	ECCUNLCK and ECCCTL[1:0] cannot be written

Bits 5:4 - ECCCTL[1:0] Condition: {PFM_PNL_DATA_MSB==71} 10 (ECC) = Reserved - Resolves to STRICT 01 (DYNAMIC) = Reserved - Resolves to STRICT Note: This field can only be modified when ECCUNLCK=1 Restricts one or more NVMOPs:

Value	Name	Description
3	DISABLE	ECC Writes with No Error Check Reads
0	STRICT	ECC Writes with ECC Reads
3	DISABLE	Dynamic Writes with No Error Check Reads
2	ECC	ECC Writes with Dynamic Reads (NVMOP = Single Program Operation disabled)
1	DYNAMIC	Dynamic Writes with Dynamic Reads
0	STRICT	ECC Writes with ECC Reads (NVMOP = Single Program Operation disabled)

31.6.10 CRC Control REGISTER

Name: CRCCTRL
Offset: 0x24
Reset: 0x00000000
Property: R/W

Table 31-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			RUNSTDBY				CRCEN	CRCRST
Access			R/W				R/W/HC	R/S/HC
Reset			0				0	0

Bit 5 – RUNSTDBY CRC Run in Standby

Value	Name	Description
1	RUN	CRC Runs in Standby but only if STATUS.PRM=0 (i.e. Flash is in Auto Standby)
0	HALT	CRC Halts in Standby

Bit 1 – CRCEN When CRCEN = 1, any write to a CRC registers labeled as Enable-Protected is discarded and an error is generated. Note: Hardware clears this bit when CRC operation finishes (i.e. when HW sets INTFLAG.CRCDONE=1).

Note: When CRCEN = 1 all other CRC* SFR bits are write protected, except CRCEN, CRCRST, and CRCPAUSE.PAUSE

Value	Name	Description
1	ENABLE	CRC Calculation Enabled (w/ start on write to 1).
0	DISABLE	Stops CRC calculation.

Bit 0 – CRCRST CRC Reset

Value	Name	Description
1	RESET	Resets all CRC SFR bits.
0	NONE	No Action

31.6.11 CRC MODE REGISTER

Name: CRCMODE
Offset: 0x28
Reset: 0x00000000
Property: R/W

Table 31-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						PERIOD[11:8]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
						PERIOD[7:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RIN	ROUT	AUTOR	PLEN32				
Access	R/W	R/W	R/W	R/W				
Reset	0	0	0	0				
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 27:16 – PERIOD[11:0] The number of PerCLK counts CRC logic waits between needing new data and reading that data from flash. 0x000 = Read Data Immediately 0x001 = Wait 1 PerCLK count + 2 AHB Clocks (for sync) ... 0xFFFF = Wait 4095 PerCLK counts + 2 AHB Clocks (for sync)

The number of PerCLK counts CRC logic waits between needing new data and reading that data from flash. (The PerCLK is clock at an 8 MHz fixed frequency. Non-zero PERIOD values are used to throttle back the bandwidth used for CRC calculations.)

0 = Read Data Immediately

All Other Values = Wait **PERIOD** PerCLK counts + 2 AHB Clocks (for sync) before starting

Bit 15 – RIN This option is sometimes referred to as Reflected Byte or Reflected Input.
This option is sometimes referred to as Reflected Byte or Reflected Input.

Value	Name	Description
1	ENABLE	The LFSR CRC is calculated Least Significant Bit first (Reflected)
0	DISABLE	The LFSR CRC is calculated Most Significant Bit first (Not Reflected)

Bit 14 – ROUT This option is sometimes referred to as Reflected Result or Reflected Output.
This option is sometimes referred to as Reflected Result or Reflected Output.

Value	Name	Description
1	ENABLE	The CRCACC is Reflected (before the Final XOR)

Value	Name	Description
0	DISABLE	The CRCACC is Not Reflected

Bit 13 – **AUTOR** CRC Auto Repeat

Value	Name	Description
1	ENABLE	Continually Repeat CRC calculation; stop on error, set CRCDONE and CRCERR
0	DISABLE	Perform CRC calculation once then set DONE and if needed, CRCERR.

Bit 12 – **PLEN32** Polynomial Length Select

Value	Name	Description
1	THIRTYTWO	Polynomial is 32-bits
0	SIXTEEN	Polynomial is 16-bits

31.6.12 CRC Pause REGISTER

Name: CRCPAUSE
Offset: 0x2c
Reset: 0x00000000
Property: R/W

Table 31-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								PAUSE
Access								R/W
Reset								0

Bit 0 – PAUSE Prevent the CRC FSM from reading flash memory so not to interfere with CPU activity. Note: The CRC calculation continues until it needs more data, and then it pauses.

Note: The CRC calculation continues until it needs more data, and then it pauses.

Prevent the CRC FSM from reading Flash memory so as to not interfere with CPU activity:

Value	Name	Description
1	ENABLE	Pause CRC Reads of Flash
0	DISABLE	CRC Reads Flash as Required

31.6.13 CRC Message Address REGISTER

Name: CRCMADR
Offset: 0x30
Reset: 0x00000000
Property: R/W

Table 31-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					CRCMADR[27:24]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCMADR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCMADR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCMADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:0 – CRCMADR[27:0] This is the system physical address of the first byte in the message. Note: Not all bits in this register are writable. CRCMADR[27:24] determines which flash region (CFM, BFM, PFM) is targeted and always exist. For CRCMADR[23:0] only the number of bits required to address the largest flash region (PFM) exists. Bits that don't exist are treated as R-0. The R/W bits of [23:0] match the bits for CRCMLEN.

This is the system physical address of the first byte of the message.

31.6.14 CRC Message Length REGISTER

Name: CRCMLLEN
Offset: 0x34
Reset: 0x00000000
Property: R/W

Table 31-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	CRCMLLEN[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCMLLEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCMLLEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – CRCMLLEN[23:0] Note: Software should limit the value written to CRCMLLEN such that CRCMADR + CRCMLLEN does not exceed the end of the flash region (i.e. BFM, PFM, or on a secure device between security regions) containing CRCMADR.

Note: Only the number of bits needed to address the largest contiguous Flash region are implemented.

31.6.15 CRC Initial Value REGISTER

Name: CRCIV
Offset: 0x38
Reset: 0x00000000
Property: R/W

Table 31-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCIV[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCIV[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCIV[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCIV[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCIV[31:0] CRCIV is loaded into CRCACC before the start of each CRC calculation, including at the start of repeat loops. When CRCMODE.PLEN32==0: CRCIV[31:16] is don't care. CRCIV[15:0] is the initial value. When CRCMODE.PLEN32==1: CRCIV[31:0] is the initial value.

CRCIV is loaded into CRCACC before the start of each CRC calculation, including at the start of repeat loops.

When CRCMODE.PLEN32=0:

CRCIV[31:16] is don't care.

CRCIV[15:0] is the initial value.

When CRCMODE.PLEN32=1:

CRCIV[31:0] is the initial value.

31.6.16 CRC Accumulator REGISTER

Name: CRCACC
Offset: 0x3c
Reset: 0x00000000
Property: R

Table 31-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCACC[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCACC[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCACC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCACC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCACC[31:0] When CRCMODE.PLEN32==0: CRCACC[31:16] is don't care. CRCACC[15:0] is the accumulator. When CRCMODE.PLEN32==1: CRCACC[31:0] is the accumulator.
When CRCMODE.PLEN32=0: CRCACC[31:16] is don't care. CRCACC[15:0] is the accumulator.
When CRCMODE.PLEN32=1: CRCACC[31:0] is the accumulator.

31.6.17 CRC Polynomial REGISTER

Name: CRCPOLY
Offset: 0x40
Reset: 0x00000001
Property: R/W

Table 31-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCPOLY[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCPOLY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCPOLY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCPOLY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	1

Bits 31:0 – CRCPOLY[31:0] When CRCMODE.PLEN32==0: CRCPOLY[31:16] is don't care. CRCPOLY[15:0] is the polynomial coefficients. CRCPOLY[0] and CRCPOLY[16] are implied and are always 1. When CRCMODE.PLEN32==1: CRCPOLY[31:0] is the polynomial coefficients. CRCPOLY[0] and CRCPOLY[32] are implied and are always 1.

When CRCMODE.PLEN32=0: CRCPOLY[31:16] is don't care. CRCPOLY[15:0] is the polynomial coefficients.

CRCPOLY[0] and CRCPOLY[16] are implied and are always 1.

When CRCMODE.PLEN32=1: CRCPOLY[31:0] is the polynomial coefficients.

CRCPOLY[0] and CRCPOLY[32] are implied and are always 1.

Value	Name	Description
1		Enable the XOR input to the shift register at the associated bit position
0		Disable the XOR input to the shift register at the associated bit position

31.6.18 CRC Final XOR REGISTER

Name: CRCFXOR
Offset: 0x44
Reset: 0x00000000
Property: R/W

Table 31-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCFXOR[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCFXOR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCFXOR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCFXOR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCFXOR[31:0] CRCSUM is compared to the XOR of CRCACC and CRCFXOR. When CRCMODE.PLEN32==0: CRCFXOR[31:16] is don't care. CRCFXOR[15:0] is the Final XOR. When CRCMODE.PLEN32==1: CRCFXOR[31:0] is the Final XOR.

CRCSUM is compared to the XOR of CRCACC and CRCFXOR.

When CRCMODE.PLEN32=0: CRCFXOR[31:16] is don't care. CRCFXOR[15:0] is the Final XOR.

When CRCMODE.PLEN32=1: CRCFXOR[31:0] is the Final XOR.

Value	Name	Description
1		Enable the Final XOR at the associated bit position
0		Disable the Final XOR at the associated bit position

31.6.19 CRC CheckSUM REGISTER

Name: CRCSUM
Offset: 0x48
Reset: 0x00000000
Property: R/W

Table 31-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CRCSUM[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CRCSUM[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRCSUM[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CRCSUM[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CRCSUM[31:0] CRCSUM is compared to the XOR of CRCACC and CRCFXOR after CRCMLen bytes have been computed. When CRCMODE.PLEN32==0: CRCFXOR[31:16] is don't care. CRCFXOR[15:0] is the CRC checksum. When CRCMODE.PLEN32==1: CRCFXOR[31:0] is the CRC checksum.

CRCSUM is compared to the XOR of CRCACC and CRCFXOR after CRCMLen bytes have been computed.

When CRCMODE.PLEN32=0: CRCFXOR[31:16] is don't care. CRCFXOR[15:0] is the CRC checksum.

When CRCMODE.PLEN32=1: CRCFXOR[31:0] is the CRC checksum.

31.6.20 Flash ECC Fault Control REGISTER

Name: FFLTCTRL
Offset: 0x4c
Reset: 0x00000000
Property: R/W

Table 31-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							FLTEN	FLTRST
Access							R/W	R/S/HC
Reset							0	0

Bit 1 – FLTEN When FLTEN = 1, any write to a FFLT registers labeled as Enable-Protected is discarded and an error is generated.

When FLTEN = 1, any write to a FFLT registers labeled as Enable-Protected is discarded and an error is generated.

Value	Name	Description
1	ENABLE	ECC Fault Injection Enabled (module performs operation selected by FFLTMODE.FLTMD)
0	DISABLE	ECC Fault Injection Disabled

Bit 0 – FLTRST Fault Reset

Value	Name	Description
1	RESET	Resets all FLT SFR bits.
0	NONE	No Action

31.6.21 Flash ECC Fault Mode REGISTER

Name: FFLTMODE
Offset: 0x50
Reset: 0x00000000
Property: R/W

Table 31-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			FLTMD[2:0]				CTLFLT[2:0]	
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 14:12 – FLTMD[2:0] Fault Mode Control

Note: Write Protected when FLTEN = 1.

000 = Fault Injection Disabled
001 = Reserved
010 = Fault Capture Mode Enabled
Capture the address (in FLTADR) and Syndrome in (FLTSYN)
011 =Reserved
100 = Single Fault Injection (at bit selected by FLT1PTR) for Reads
101 = Double Fault Injection for Reads
110 = Single Fault Injection (at bit selected by FLT1PTR) for Writes
111 = Double Fault Injection for Writes

Value	Name	Description
0	DISABLE	Fault Injection Disabled
2	CAPT	Fault Capture Mode Enabled - Captures the address (in FLTADR) and Syndrome in (FLTSYN)
4	SINGLEREAD	Single Fault Injection (at bit selected by FLT1PTR) for Reads
5	DOUBLEREAD	Double Fault Injection for Reads
6	SINGLEWRITE	Single Fault Injection (at bit selected by FLT1PTR) for Writes
7	DOUBLEWRITE	Double Fault Injection for Writes

Bits 10:8 – CTLFLT[2:0] If Fault Injection Mode, this defines the bit on which a fault is injected for CTL.

Note: Write Protected when FLTEN = 1.

If FLTMD = 1xx and FLTEN = 1:

Value	Name	Description
1		Inject a Fault on to the associated ECC/Parity Control bits (CTL[n])
0		No Fault Injected

31.6.22 Flash ECC Fault Pointer REGISTER

Name: FFLTPTR
Offset: 0x54
Reset: 0x00000000
Property: R/W

Table 31-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
								FLTPTR2[8]
Access								R/W
Reset								0
Bit	23	22	21	20	19	18	17	16
	FLTPTR2[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
								FLTPTR1[8]
Access								R/W
Reset								0
Bit	7	6	5	4	3	2	1	0
	FLTPTR1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0:8, 16:24 – FLTPTR If Fault Injection Mode, this is the pointer to the bit in the parity calculation vector on which to generate a fault. Condition: {PFM_DATA_MSB==63} 11111111 ... 001001000 = Don't Use - UNDEFINED RESULT (above max bit) 001000111 = Inject fault on Bit 71 (max bit for 128-bit data + 8-bit Parity) ... 000000001 = Inject fault on Bit 0 000000000 = Inject fault on Bit 0 Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Condition: {PFM_DATA_MSB==127} 11111111 ... 010001001 = Don't Use - UNDEFINED RESULT (above max bit) 010001000 = Inject fault on Bit 136 (max bit for 128-bit data + 9-bit Parity) ... 000000001 = Inject fault on Bit 1 000000000 = Inject fault on Bit 0 Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Condition: {PFM_DATA_MSB==255} 11111111 ... 100001010 = Don't Use - UNDEFINED RESULT (above max bit) 100001001 = Inject fault on Bit 265 (max bit for 256-bit data + 10-bit Parity) ... 000000001 = Inject fault on Bit 1 000000000 = Inject fault on Bit 0 Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order.

31.6.23 Flash ECC Fault Address REGISTER

Name: FFLTADR
Offset: 0x58
Reset: 0x00000000
Property: R/W

Table 31-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					FLTADR[27:24]			
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FLTADR[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FLTADR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FLTADR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 27:0 – FLTADR[27:0] If Fault Injection Mode, this is the System Physical Address at which to inject fault(s)
Note: Not all bits in this register are writable. FFLTADR[27:24] determines which flash region (CFM, BFM, PFM) is targeted and always exist. For FFLTADR[23:0] only the number of bits required to address the largest flash region (PFM) exists. Bits that don't exist are treated as R-0. Note: If FLTADR is within VSS Page space, FLT logic is disabled. Condition: {PFM_DATA_MSB==63} This is a byte address but aligned to 8 bytes and FLTADR[2:0] are always 0. Condition: {PFM_DATA_MSB==127} This is a byte address but aligned to 16 bytes and FLTADR[3:0] are always 0. Condition: {PFM_DATA_MSB==255} This is a byte address but aligned to 32 bytes and FLTADR[4:0] are always 0.

In Fault Injection Mode this is the System Physical Address at which to inject fault(s).

Notes:

1. This is a byte address but aligned to 16 bytes and FLTADR[3:0] are always 0.
2. If FLTADR is within VSS Page space, FLT logic is disabled.

31.6.24 Flash ECC Fault Capture Address REGISTER

Name: FFLTCAP
Offset: 0x5c
Reset: 0x00000000
Property: R

Table 31-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					FLTADR[27:24]			
Access					R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FLTADR[23:16]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FLTADR[15:8]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FLTADR[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bits 27:0 – FLTADR[27:0] If Fault Capture Mode, this is the Flash Physical Address at which a fault was detected. Note: Not all bits in this register exist. FFLTADR[27:24] determines which flash region (CFM, BFM, PFM) has been captured and always exist. For FFLTADR[23:0] only the number of bits required to address the largest flash region (PFM) exists. Bits that don't exist are treated as R-0. Note: FLT logic never captures addresses within the VSS Page space or any other CFM page that is Read Protected.

In Fault Capture Mode this is the Flash Physical Address at which a fault was detected.

Note: FLT logic never captures addresses within the VSS Page space or any other CFM page that is Read Protected.

31.6.25 Flash ECC Fault Parity REGISTER

Name: FFLTPAR
Offset: 0x60
Reset: 0x00000000
Property: R

Table 31-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DEDOUT							SECOUT[8]
Access	R/HS/HC							R/HS/HC
Reset	0							0
Bit	23	22	21	20	19	18	17	16
	SECOUT[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DEDIN							SECIN[8]
Access	R/HS/HC							R/HS/HC
Reset	0							0
Bit	7	6	5	4	3	2	1	0
	SECIN[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bit 31 – DEDOUT For Flash Writes this value is based on write data and the calculated SEC Parity bits. For Flash Reads this value is based on read data and the calculated SEC Parity bits. Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Note: This document refers to DED, Overall Parity, and Parity[0] interchangeably.

For Writes this value is based on write data and the calculated SEC Parity bits.

For Reads this value is based on read data and the calculated SEC Parity bits.

Notes:

1. See the table for calculation vector bit order vs data bit order vs control bit order.
2. “DED”, “Overall Parity”, and “Parity[0]” are used interchangeably.

Bits 24:16 – SECOUT[8:0] For Flash Writes this value is based on write data. For Flash Reads this value is based on read data. Condition: {PFM_DATA_MSB==63} SECOUT[8:7] are always 0. Condition: {PFM_DATA_MSB==127} SECOUT[8] is always 0. Condition: {PFM_DATA_MSB==255} All SECOUT bits are used. Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Note: The number of active bits is dependent on the data width of the flash panel. Note: This document refers to SECOUT[8:0] and Parity[9:1] interchangeably.

For Writes this value is based on write data.

For Reads this value is based on read data.

Notes:

1. See the table for calculation vector bit order vs data bit order vs control bit order.
2. SECOUT[8].
3. The terms “SEC*[8:0]” and “Parity[9:1]” are used interchangeably.

Bit 15 – DEDIN For Flash Writes this value is always 0. For Flash Reads this value is the overall parity read from flash. Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Note: This document refers to DED, Overall Parity, and Parity[0] interchangeably.

For Writes this value is always 0.

For Reads this value is the overall parity read from flash.

Notes:

1. See the table for calculation vector bit order vs data bit order vs control bit order.
2. “DED”, “Overall Parity”, and “Parity[0]” are used interchangeably.

Bits 8:0 – SECIN[8:0] For Flash Writes this value is always 0. For Flash Reads this value is the Single Error Parity bits read from flash. Condition: {PFM_DATA_MSB==63} SECIN[8:7] are always 0. Condition: {PFM_DATA_MSB==127} SECIN[8] is always 0. Condition: {PFM_DATA_MSB==255} All SECIN bits are valid. Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Note: The number of active bits is dependent on the data width of the flash panel. Note: This document refers to SECIN[8:0] and Parity[9:1] interchangeably.

For Writes this value is always 0.

For Reads this value is the Single Error Parity bits read from Flash.

Notes:

1. See the table for calculation vector bit order vs data bit order vs control bit order.
2. SECIN[8] is always 0.
3. The terms “SEC*[8:0]” and “Parity[9:1]” are used interchangeably.

31.6.26 Flash ECC Fault Syndrome REGISTER

Name: FFLTSYN
Offset: 0x64
Reset: 0x00000000
Property: R

Table 31-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERR[3:0]					CTLSTAT[2:0]		
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC		R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0		0	0	0
Bit	23	22	21	20	19	18	17	16
						CERR	DSERR[1:0]	
Access						R/HS/HC	R/HS/HC	R/HS/HC
Reset						0	0	0
Bit	15	14	13	12	11	10	9	8
	DEDSYN							SECSYN[8]
Access	R/HS/HC							R/HS/HC
Reset	0							0
Bit	7	6	5	4	3	2	1	0
	SECSYN[7:0]							
Access	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC	R/HS/HC
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – PERR[3:0] Note: Word size is defined by the Write Word size of the flash, either 32-bits or 64-bits.

Value	Name	Description
1	ERROR	Parity Error on Word n
0	NONE	No Parity Error on Word n

Bits 26:24 – CTLSTAT[2:0] Note: Panel is always programmed with 000 for ECC and 111 for parity.

Note: Panel is always programmed with 000 for ECC and 111 for parity.

000,001,010,100 = Calculation used ECC (i.e. programming used quad write)

011,101,110,111 = Calculation used Parity (i.e. programming used single write)

Value	Name	Description
0	USEDECC_0	Calculation used ECC (i.e. programming used quad write)
1	USEDECC_1	Calculation used ECC (i.e. programming used quad write)
2	USEDECC_10	Calculation used ECC (i.e. programming used quad write)
4	USEDECC_100	Calculation used ECC (i.e. programming used quad write)
3	USEDPARITY_11	Calculation used Parity (i.e. programming used single write)
5	USEDPARITY_101	Calculation used Parity (i.e. programming used single write)
6	USEDPARITY_110	Calculation used Parity (i.e. programming used single write)
7	USEDPARITY_111	Calculation used Parity (i.e. programming used single write)

Bit 18 – CERR ECC Control bit Error

Value	Name	Description
1	SINGLE	Single Control Bit Error
0	NONE	No Control bit Error (ECCSTAT either 111 or 000)

Bits 17:16 – DSERR[1:0] Evaluates as [DERR,SERR] For Reads only when ECCSTAT = ECC
For Reads only when ECCSTAT = ECC

Value	Name	Description
0	NONE	No Errors
1	SINGLE	Single Error Corrected
2	DOUBLE_10	Double Error Detected
3	DOUBLE_11	Double Error Detected

Bit 15 – DEDSYN This is Overall Parity Calculated from Data and all Parity bits read from flash.
This is Overall Parity Calculated from Data and all Parity bits read from Flash.

Value	Name	Description
1		Calculated Overall Parity Differs from Read Overall Parity
0		Calculated Overall Parity Concurs with Read Overall Parity

Bits 8:0 – SECSYN[8:0] For Reads only when CTLSTAT = ECC or System bits ECCCTL[1:0]=ECC This value is the bitwise XOR of SECIN and SECOUT. When FFLTSYN.DEDSYN==1: 000000000 = No Data Error, but DED bit in Error Others = SECSYN[] points to the bit position in the calculation vector that was corrected When FFLTSYN.DEDSYN==0: 000000000 = No Data Error and No DED bit Error Others = Double Error Detected.
Note: The number of active bits is dependent on the data width of the flash panel.

For Reads only when CTLSTAT = ECC or System bits ECCCTL[1:0]=ECC
This value is the bitwise XOR of SECIN and SECOUT.

If DEDSYN=1:

000000000 = No Data Error, but DED bit in Error

Non-Zero = SECSYN points to the bit position in the calculation vector that was corrected

If DEDSYN = 0:

000000000 = No Data Error or DED bit Error

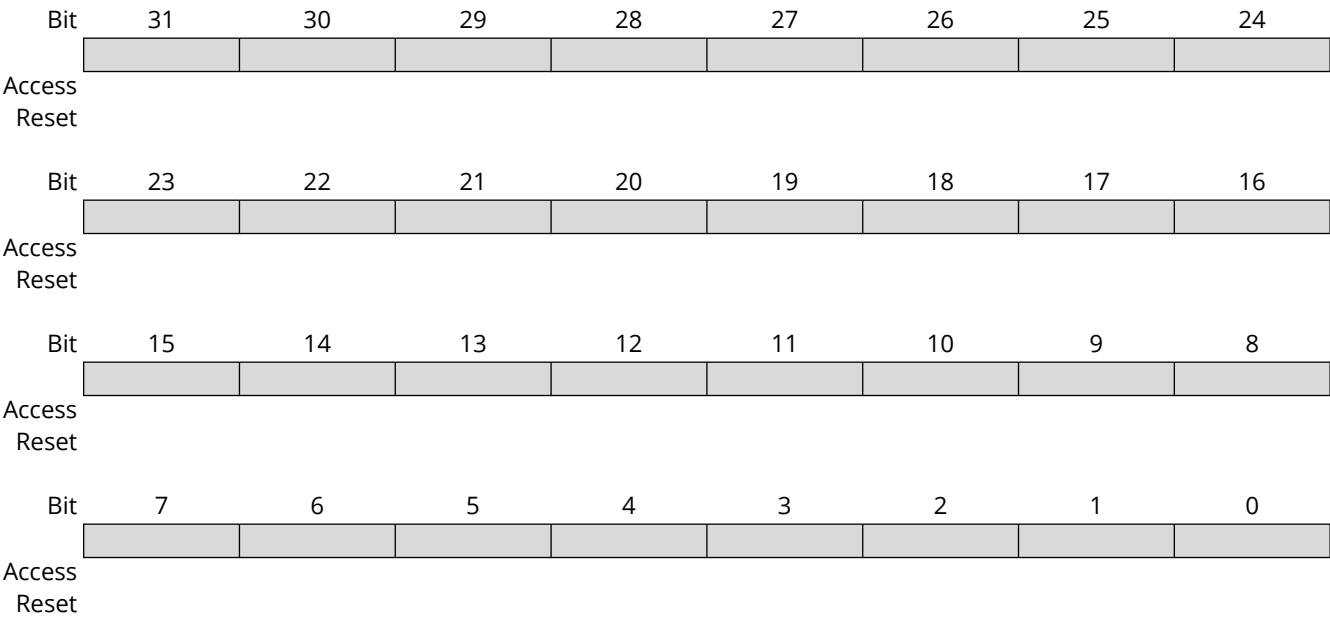
Non-Zero = Double Error Detected.

31.6.27 CFM Page Read Protection REGISTER (CFM8)

Name: CRP
Offset: 0x68
Reset: 0x000C0000
Property: R/W

Table 31-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—



31.6.28 Write Protection Control REGISTER

Name: WPCTRL
Offset: 0x70
Reset: 0x00000000
Property: R/W

Table 31-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
					WPFFLTEN	WPCRCEN	WPLCK	WPEN
Access					R/W	R/W	R/S	R/W
Reset					0	0	0	0

Bits 31:8 – WPKEY[23:0] Writing a value other than FCR_WPCTRL_KEY to this field cancels the write operation this register and generates a slave bus error. This field always returns 0 on read.

Bit 3 – WPFFLTEN Note: FFLTMODE.FLTMD[] selection 001 and 011 which lock the mode will eventually be de-featured since they are redundant with the WPCTRL.WPFFLTEN feature.

Value	Name	Description
1	ENABLE	FFLT Register write protection enabled. Non-debugger writes to FLT registers marked with write protect property are canceled and generate a slave bus error.
0	DISABLE	FFLT Register write protection disabled.

Bit 2 – WPCRCEN Write Protection Enable for CRC Registers

Value	Name	Description
1	ENABLE	CRC Register write protection enabled. Non-debugger writes to CRC registers marked with write protect property are canceled and generate a slave bus error.
0	DISABLE	CRC Register write protection disabled.

Bit 1 – WPLCK WPCTRL Write LOCK

Value	Name	Description
1	LOCKED	WPCTRL register is write protected. This bit can only be cleared by a hardware reset.

Value	Name	Description
0	UNLOCKED	WPCTRL is not write protected.

Bit 0 – WPEN Note: Write Protection also applies to HSM accesses for this module.

Value	Name	Description
1	ENABLE	Register write protection enabled (for registers not associated with a WP<feature>EN bit). Non-debugger writes to registers marked with write protect property are canceled and generate a slave bus error.
0	DISABLE	Register write protection disabled.

32. Event System (EVSYS)

32.1 Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals. Several peripherals can be configured to generate and respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have many event generators and can have multiple event users.

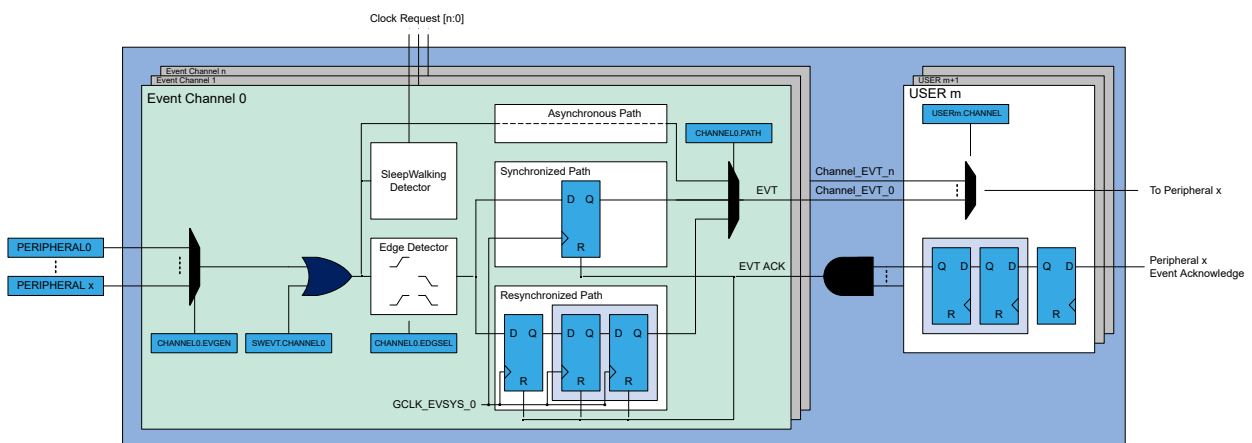
Communication is made without CPU intervention and without consuming system resources, such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

32.2 Features

- 12 configurable event channels:
 - All channels can be connected to any event generator
 - All channels provide a pure asynchronous path
 - All channels provide a resynchronized path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- Device-specific event generators
- Device-specific event users
- Configurable edge detector
- Peripherals can be event generators, event users, or both
- SleepWalking and interrupt for operation in sleep modes
- Software event generation
- Each event user can choose which channel to respond to
- Optional Static or Round-Robin interrupt priority arbitration

32.3 Block Diagram

Figure 32-1. Event System Block Diagram



32.4 Power Management

The EVSYS can be used to wake up the CPU from a sleep mode (except BACKUP and OFF Mode), even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM - Power Manager* for details on the different sleep modes.

Although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CHANNEL_n) will be restarted if that channel uses a resynchronized path. It does not need to wake the system from sleep.



Important: This generic clock only applies to channels which can be configured asynchronous or resynchronized.

32.5 Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module (MCLK). The EVSYS APB BUS interface clocks, CLK_EVSYS_APB, are enabled by default on reset. (See *MCLK*).

Each EVSYS channel which can be configured as synchronous or resynchronized has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to the *GCLK - Generic Clock Controller* chapter for details.

32.6 Functional Description

32.6.1 Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or I/O pins (users). Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using a resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on event generators.

For further details, refer to the Channel Path section of this chapter.

32.6.2 Basic Operation

32.6.2.1 Initialization

Before enabling event routing within the system, the Event Users Multiplexer and Event Channels must be selected in the Event System (EVSYS), and the two peripherals that generate and use the event have to be configured. The recommended sequence is:

1. In the event generator peripheral, enable output of event by writing a '1' to the respective Event Output Enable bit ("EO") in the peripheral's Event Control register (i.e., TCC.EVCTRL.MCEO1, AC.EVCTRL.WINEO0, RTC.EVCTRL.OVFEO).
2. Configure the EVSYS:
 - a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, see User Multiplexer Setup.
 - b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, see Event System Channel.

3. Configure the action to be executed by the event user peripheral by writing to the Event Action bits (EVACT) in the respective Event control register (i.e., TCC.EVCTRL.EVACT).
Note: Not all peripherals require this step.
4. In the event user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register (i.e., AC.EVCTRL.IVEIO, ADC.EVCTRL.STARTEI).

32.6.2.2 User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Block Diagram section. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all user multiplexers is found in the User (USERm) register description.

32.6.2.3 Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be resynchronized or asynchronously sent to the users. When resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands, for additional information, refer to the [Figure 32-1](#).

32.6.2.4 Event Generators

Each event channel can receive the events from all event generators. All event generators are listed in the [Peripherals Configuration Summary](#) and configured in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0).

32.6.2.5 Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

Asynchronous Path

When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel (GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel x Status register (CHSTATUSx) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

Resynchronized Path

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel.

When the resynchronized path is used, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

32.6.2.6 Edge Detection

When resynchronized path is are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

32.6.2.7 Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel configuration.

- **Asynchronous Path:** The maximum routing latency of an external event is related to the internal signal routing and it is device dependent
- **Resynchronized Path:** The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles. Channel status and interrupt flags are set when the event routing is complete.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

32.6.2.8 The Overrun Channel n Interrupt

The Overrun Channel n Interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.OVR) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event
- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the CHINTFLAGn.OVR is always read as zero.

32.6.2.9 The Event Detected Channel n Interrupt

The Event Detected Channel n Interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.EVD) is set when an event coming from the event generator configured on channel n is detected.

The flag will only be set when using a resynchronized path. In the case of an asynchronous path, the CHINTFLAGn.EVD is always zero.

32.6.2.10 Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUSn.BUSYCH bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel
- The CHSTATUSn.RDYUSR bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel

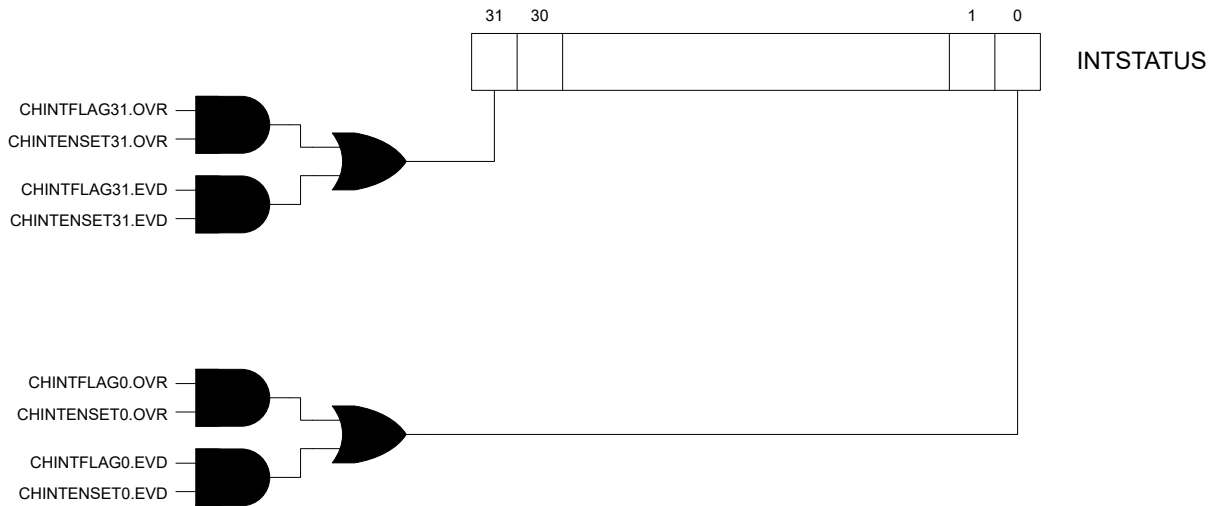
32.6.2.11 Software Event

A software event can be initiated on a channel by writing a '1' to the Software Event bit in the Channel register (CHANNELm.SWEVT). Then the software event can be serviced as any event generator; i.e., when the bit is set to '1', an event will be generated on the respective channel.

32.6.2.12 Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as shown below.

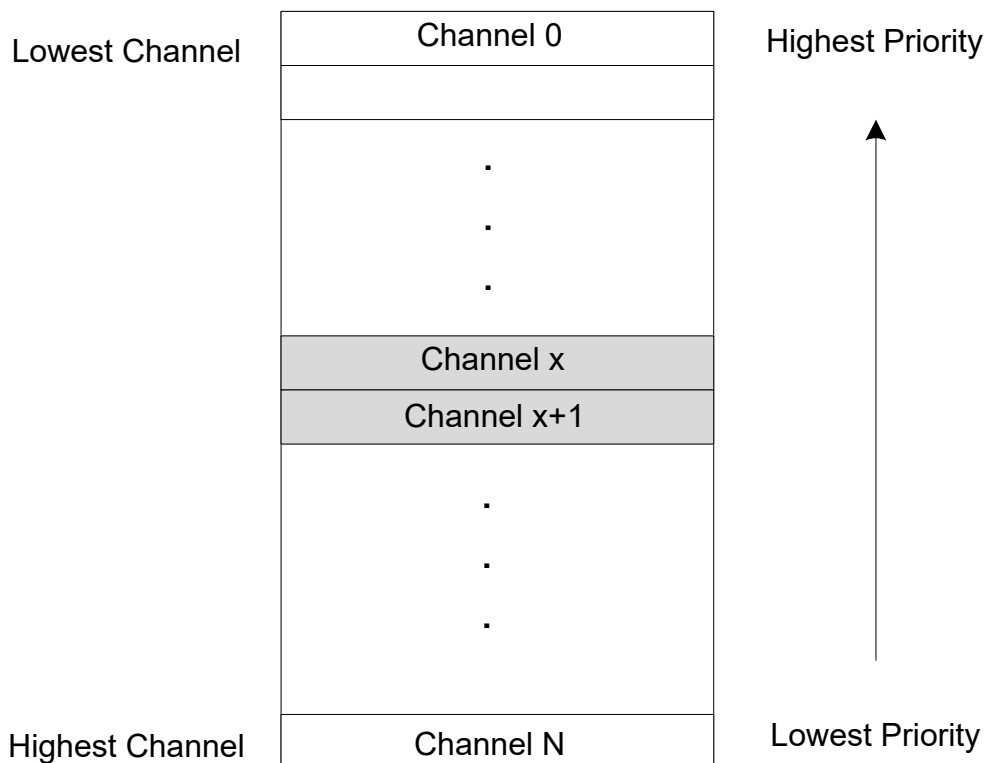
Figure 32-2. Interrupt Status Register



The Event System can arbitrate between all channels with pending interrupts. The arbiter can be configured to prioritize statically or dynamically the incoming events. The priority is evaluated each time a new channel has an interrupt pending, or an interrupt has been cleared. The Channel Pending Interrupt register (INTPEND) will provide the channel number with the highest interrupt priority, and the corresponding channel interrupt flags and status bits.

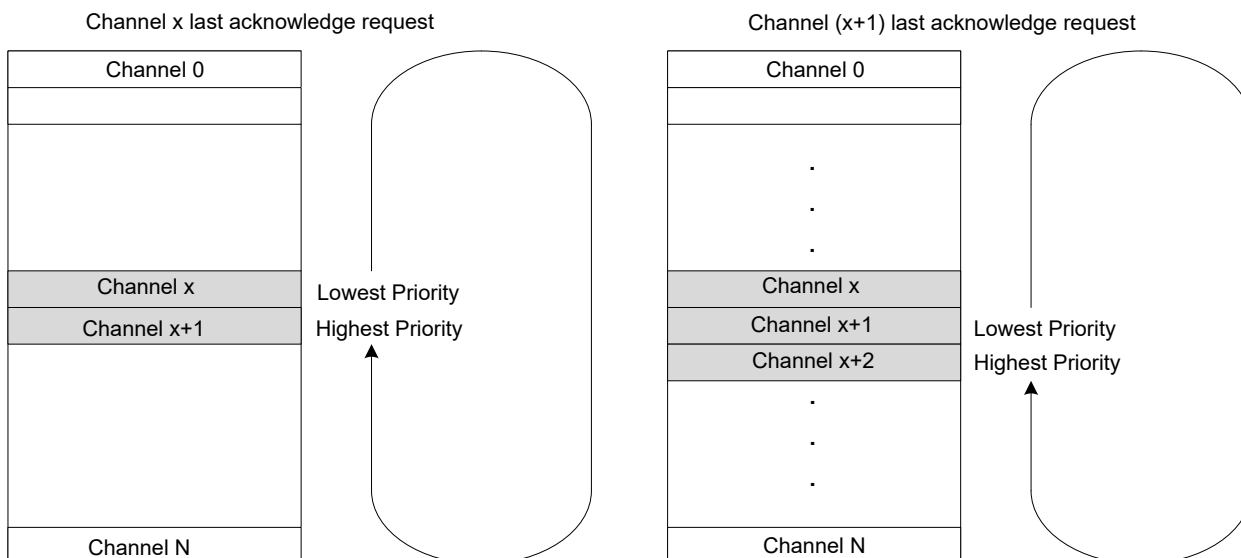
By default, static arbitration is enabled (PRICTRL.RRENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown below. When using the status scheme, there is a risk of high channel numbers never being granted access by the arbiter. This can be avoided using a dynamic arbitration scheme.

Figure 32-3. Static Priority



The dynamic arbitration scheme available in the Event System is round-robin. Round-robin arbitration is enabled by writing `PRICTRL.RREN` to one. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel, as shown below. The channel number of the last channel being granted access, will be stored in the Channel Priority Number bit group in the Priority Control register (`PRICTRL.PRI`).

Figure 32-4. Round-Robin Scheduling



The Channel Pending Interrupt register (INTPEND) also offers the possibility to indirectly clear the interrupt flags of a specific channel. Writing a flag to one in this register, will clear the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

32.6.3 Interrupts

The EVSYS has the following interrupt sources for each channel:

- Overrun Channel n interrupt (OVR)
- Event Detected Channel n interrupt (EVD)

These interrupts events are asynchronous wake-up sources.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the corresponding Channel n Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs.

Note: Interrupts must be globally enabled to allow the generation of interrupt requests.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Set (CHINTENSET) register, and disabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Clear (CHINTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the Event System is reset. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts, and must read the Channel n Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the highest priority channel with pending interrupt and the respective interrupt flags.

32.6.4 Sleep Mode Operation

The Event System can generate interrupts to wake up the device from IDLE or STANDBY sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in Channel register

(CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND:

Table 32-1. Event Channel Sleep Behavior

CHANNELn.PATH	CHANNELn.ONDEMAND	CHANNELn.RUNSTDBY	Sleep Behavior
ASync	0	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode.
RESync	0	1	Run in both IDLE and STANDBY sleep modes.
RESync	1	0	Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode. Two GCLK_EVSYS_n latency added in RESync path before the event is propagated internally.
RESync	1	1	Run in both IDLE and STANDBY sleep modes. Two GCLK_EVSYS_n latency added in RESync path before the event is propagated internally.

32.7 Peripheral Dependencies

Table 32-2. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
EVSYS	0x44806000	48 : EVD0/ OVR0 49 : EVD1/ OVR1 50 : EVD2/ OVR2 51 : EVD3/ OVR3 52 : EVD4/ OVR4 53 : EVD5/ OVR5 54 : EVD6/ OVR6 55 : EVD7/ OVR7 56 : EVD8/ OVR8 57 : EVD9/ OVR9 58 : EVD10/ OVR10 59 : EVD11/ OVR11	APB: CLKMSK[1] MASK3	CH0: PCHCTRL[8] CH1: PCHCTRL[9] CH2: PCHCTRL[10] CH3: PCHCTRL[11] CH4: PCHCTRL[12] CH5: PCHCTRL[13] CH6: PCHCTRL[14] CH7: PCHCTRL[15] CH8: PCHCTRL[16] CH9: PCHCTRL[17] CH10: PCHCTRL[18] CH11: PCHCTRL[19]	25 STATUS0 PERID25			PD_CORE_SW

32.8 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0								SWRST
0x01	Reserved									
...										
0x03										
0x04	SWEVT	31:24	CHANNEL[31:24]							
		23:16	CHANNEL[23:16]							
		15:8	CHANNEL[15:8]							
		7:0	CHANNEL[7:0]							
0x08	PRICTRL	7:0	RREN					PRI[4:0]		
0x09	Reserved									
...										
0x0F										
0x10	INTPEND	15:8	BUSY	READY					EVD	OVR
		7:0				ID[4:0]				
0x12	Reserved									
...										
0x13										
0x14	INTSTATUS	31:24	CHINT[31:24]							
		23:16	CHINT[23:16]							
		15:8	CHINT[15:8]							
		7:0	CHINT[7:0]							
0x18	BUSYCH	31:24	BUSYCH[31:24]							
		23:16	BUSYCH[23:16]							
		15:8	BUSYCH[15:8]							
		7:0	BUSYCH[7:0]							
0x1C	READYUSR	31:24	READYUSR[31:24]							
		23:16	READYUSR[23:16]							
		15:8	READYUSR[15:8]							
		7:0	READYUSR[7:0]							
0x20	CHANNEL	31:24								
		23:16								
		15:8	ONDEMAND	RUNSTDBY				EDGSEL[1:0]		PATH[1:0]
		7:0	EVGEN[7:0]							
0x24	CHINTENCLR	7:0							EVD	OVR
0x25	CHINTENSET	7:0							EVD	OVR
0x26	CHINTFLAG	7:0							EVD	OVR
0x27	CHSTATUS	7:0							BUSYCH	RDYUSR
0x28	Reserved									
...										
0x011F										
0x0120	USER0	7:0	CHANNEL[7:0]							
...										
0x0133	USER19	7:0	CHANNEL[7:0]							
0x0134	Reserved									
...										
0x01D3										
0x01D4	INTENCLR	7:0								NSCHK
0x01D5	INTENSET	7:0								NSCHK
0x01D6	INTFLAG	7:0								NSCHK
0x01D7	Reserved									
0x01D8	NONSECCHAN	31:24	CHANNEL[31:24]							
		23:16	CHANNEL[23:16]							
		15:8	CHANNEL[15:8]							
		7:0	CHANNEL[7:0]							
0x01DC	NSCHKCHAN	31:24	CHANNEL[31:24]							
		23:16	CHANNEL[23:16]							
		15:8	CHANNEL[15:8]							
		7:0	CHANNEL[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01E0	NONSECUSER	31:24	USER[31:24]							
		23:16	USER[23:16]							
		15:8	USER[15:8]							
		7:0	USER[7:0]							
0x01E4 ... 0x01EF	Reserved									
0x01F0	NSCHKUSER	31:24	USER[31:24]							
		23:16	USER[23:16]							
		15:8	USER[15:8]							
		7:0	USER[7:0]							

32.8.1 Control

Name: CTRLA
Offset: 0x000
Reset: 0x00
Property: W

Table 32-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								SWRST
Access								W
Reset								0

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the EVSYS to their initial state. It will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Note: Before applying a Software Reset it is recommended to disable the event generators.

32.8.2 Software Event

Name: SWEVT
Offset: 0x004
Reset: 0x00000000
Property: W

Table 32-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHANNEL[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHANNEL[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHANNEL[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHANNEL[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHANNEL[31:0] Channel x Software Selection

32.8.3 Priority Control

Name: PRICTRL
Offset: 0x008
Reset: 0x00
Property: RW

Table 32-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	RREN			PRI[4:0]				
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – RREN Round-Robin Scheduling Enable

For details on scheduling schemes, refer to Interrupt Status and Interrupts Arbitration.

Value	Description
0	Static scheduling scheme for channels with level priority
1	Round-robin scheduling scheme for channels with level priority

Bits 4:0 – PRI[4:0] Channel Priority Number

When round-robin arbitration is enabled (PRICTRL.RREN=1) for priority level, this register holds the channel number of the last EVSYS channel being granted access as the active channel with priority level. The value of this bit group is updated each time the INTPEND or any of CHINTFLAG registers are written.

When static arbitration is enabled (PRICTRL.RREN=0) for priority level, and the value of this bit group is nonzero, it will not affect the static priority scheme.

This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).

32.8.4 Channel Pending Interrupt

Name: INTPEND
Offset: 0x010
Reset: 0x4000
Property: RW

An interrupt that handles several channels should consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

Table 32-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	BUSY	READY					EVD	OVR
Access	R	R					RW	RW
Reset	0	1					0	0

Bit	7	6	5	4	3	2	1	0
						ID[4:0]		
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bit 15 – BUSY Busy

This bit is read '1' when the event on a channel selected by Channel ID field (ID) has not been handled by all the event users connected to this channel.

Bit 14 – READY Ready

This bit is read '1' when all event users connected to the channel selected by Channel ID field (ID) are ready to handle incoming events on this channel.

Bit 9 – EVD Channel Event Detected

This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.

When the event channel path is asynchronous, the EVD bit will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bit 8 – OVR Channel Overrun

This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.

There are two possible overrun channel conditions:

- One or more of the event users on channel selected by Channel ID field (ID) are not ready when a new event occurs
- An event happens when the previous event on channel selected by Channel ID field (ID) has not yet been handled by all event users

When the event channel path is asynchronous, the OVR interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

Bits 4:0 – ID[4:0] Channel ID

These bits store the channel number of the highest priority.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.

32.8.5 Interrupt Status

Name: INTSTATUS
Offset: 0x014
Reset: 0x00000000
Property: R

Table 32-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHINT[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHINT[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHINT[31:0] Channel x Pending Interrupt

32.8.6 Busy Channels

Name: BUSYCH
Offset: 0x018
Reset: 0x00000000
Property: R

Table 32-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	BUSYCH[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BUSYCH[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BUSYCH[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BUSYCH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – BUSYCH[31:0] Busy Channel x

32.8.7 Ready Users

Name: READYUSR
Offset: 0x01C
Reset: 0xFFFFFFFF
Property: R

Table 32-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	READYUSR[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	READYUSR[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	READYUSR[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	READYUSR[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – READYUSR[31:0] Ready User for Channel x

This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.

This bit is cleared when at least one of the event users connected to the channel is not ready.

When the event channel n path is asynchronous, this bit is always read zero.

32.8.8 Channel n Control

Name: CHANNEL
Offset: 0x020
Reset: 0x00008000
Property: RW

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

Table 32-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ONDEMAND	RUNSTDBY			EDGSEL[1:0]		PATH[1:0]	
Access	RW	RW			RW	RW	RW	RW
Reset	1	0			0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 15 – ONDEMAND Generic Clock On Demand

This bit is used to determine whether the generic clock is requested.

This bit has no effect for channels when asynchronous path is selected or for channels with asynchronous support only.

This bit is always read zero for channels with asynchronous support only.

Value	Description
0	Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled.
1	Generic clock is requested on demand while an event is handled

Bit 14 – RUNSTDBY Run in standby

This bit is used to define the behavior during standby sleep mode, for a resynchronized channel.

This bit has no effect for channels when asynchronous path is selected or for channels with asynchronous support only.

This bit is always read zero for channels with asynchronous support only.

Value	Description
0	The channel is disabled in standby sleep mode.
1	The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND bit.

Bits 11:10 – EDGSEL[1:0] Edge Detection Selection

These bits set the type of edge detection to be used on the channel.
These bits must be written to zero when using the asynchronous path.

Value	Name	Description
0	NO_EVT_OUTPUT	No event output when using the resynchronized or synchronous path
1	RISING_EDGE	Event detection only on the rising edge of the signal from the event generator when using the resynchronized or synchronous path
2	FALLING_EDGE	Event detection only on the falling edge of the signal from the event generator when using the resynchronized or synchronous path
3	BOTH_EDGES	Event detection on rising and falling edges of the signal from the event generator when using the resynchronized or synchronous path

Bits 9:8 – PATH[1:0] Path Selection

These bits are used to choose which path will be used by the selected channel.



Important: When synchronous or resynchronized path is enabled, event inversion feature in peripherals must not be enabled (EVCTRL.xxxINV = 0).

Value	Name	Description
0x0	ASYNCHRONOUS	Asynchronous path
0x0	SYNCHRONOUS	Synchronous path
0x1	RESYNCHRONIZED	Resynchronized path
0x2	ASYNCHRONOUS	Asynchronous path

Bits 7:0 – EVGEN[7:0] Event Generator Selection

These bits are used to choose the event generator to connect to the selected channel. Refer to the table for finding out the value corresponding to each module event generator output.

32.8.9 Channel n Interrupt Enable Clear

Name: CHINTENCLR
Offset: 0x024
Reset: 0x00
Property: RW

Table 32-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Disable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Event Detected Channel Interrupt Enable bit, which disables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Disable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Channel Interrupt Enable bit, which disables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

32.8.10 Channel n Interrupt Enable Set

Name: CHINTENSET
Offset: 0x025
Reset: 0x00
Property: RW

Table 32-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Event Detected Channel Interrupt Enable bit, which enables the Event Detected Channel interrupt.

Value	Description
0	The Event Detected Channel interrupt is disabled.
1	The Event Detected Channel interrupt is enabled.

Bit 0 – OVR Channel Overrun Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overrun Channel Interrupt Enable bit, which enables the Overrun Channel interrupt.

Value	Description
0	The Overrun Channel interrupt is disabled.
1	The Overrun Channel interrupt is enabled.

32.8.11 Channel n Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x026
Reset: 0x00
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 32-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected

This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.

When the event channel path is asynchronous, the EVD interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Event Detected Channel interrupt flag.

Bit 0 – OVR Channel Overrun

This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.

There are two possible overrun channel conditions:

- One or more of the event users on the channel are not ready when a new event occurs.
- An event happens when the previous event on channel has not yet been handled by all event users.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Channel interrupt flag.

32.8.12 Channel n Status

Name: CHSTATUS
Offset: 0x027
Reset: 0x01
Property: R

Table 32-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							BUSYCH	RDYUSR
Access							R	R
Reset							0	1

Bit 1 – BUSYCH Busy Channel

This bit is cleared when channel is idle.

This bit is set if an event on channel has not been handled by all event users connected to channel.
When the event channel path is asynchronous, this bit is always read '0'.

Bit 0 – RDYUSR Ready User

This bit is cleared when at least one of the event users connected to the channel is not ready.

This bit is set when all event users connected to channel are ready to handle incoming events on the channel.

When the event channel path is asynchronous, this bit is always read '0'.

32.8.13 User Multiplexer n

Name: USER
Offset: 0x0120 + n*0x01 [n=0..19]
Reset: 0x00
Property: RW

Table 32-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CHANNEL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – CHANNEL[7:0] Channel Event Selection

These bits select channel n to connect to the event user m. Refer to the table for finding out the m value corresponding to each module event user input.

Notes:

1. A value x of this bit field selects channel n = x-1.
2. By default, a channel is asynchronous. Channelresynchronous path can be enabled if resynchronous selection is written to CHANNEL.PATH bit field.

32.8.14 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x1D4
Reset: 0x00
Property: RW

Table 32-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable

32.8.15 Interrupt Enable Set

Name: INTENSET
Offset: 0x1D5
Reset: 0x00
Property: RW

Table 32-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable

32.8.16 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x1D6
Reset: 0x00
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 32-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check

32.8.17 Channels Security Attribution

Name: NONSECCHAN
Offset: 0x1D8
Reset: 0x00000000
Property: RW

Table 32-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHANNEL[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHANNEL[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHANNEL[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHANNEL[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHANNEL[31:0] Non-Secure for Channel x

32.8.18 Non-Secure Channels Check

Name: NSCHKCHAN
Offset: 0x1DC
Reset: 0x00000000
Property: RW

Table 32-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHANNEL[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHANNEL[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHANNEL[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHANNEL[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHANNEL[31:0] Channel x to be checked as non-secured

32.8.19 Users Security Attribution

Name: NONSECUSER
Offset: 0x1E0
Reset: 0x00000000
Property: RW

Table 32-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	USER[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	USER[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	USER[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	USER[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – USER[31:0] Non-Secure for User x

32.8.20 Non-Secure Users Check

Name: NSCHKUSER
Offset: 0x1F0
Reset: 0x00000000
Property: RW

Table 32-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	USER[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	USER[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	USER[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	USER[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – USER[31:0] User x to be checked as non-secured

33. I/O Pin Controller (PORT)

33.1 Overview

The IO Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package/number of pins. Each pin may either be used for general-purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general-purpose I/O, each pin can be configured as input or output, with highly configurable driver and pull settings. On PIC32CM SG devices, if the PORT peripheral is configured as mixed-secure in the H2PB, each pin can be defined as secure or non-secured, where secured pins can only be handled by secure accesses.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of one or more pins may be changed (set, reset, or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-bit, 16-bit, or 32-bit write.

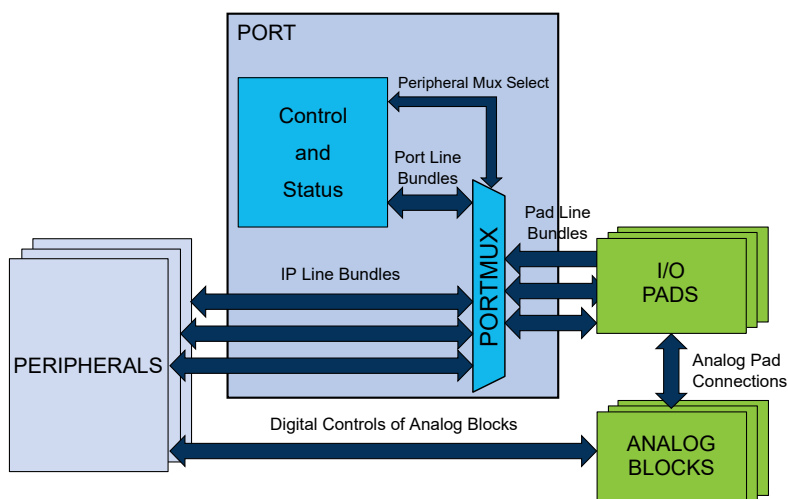
Note: If $AVDD < 2.5V$, the user must enable analog charge pumps in `SUPC.VREGCTRL.CPEN[2:0]`.

33.2 Features

- Selectable input and output configuration for each individual pin
- Software-controlled multiplexing of peripheral functions on I/O pins
- Flexible pin configuration through a dedicated Pin Configuration register
- Configurable output driver:
 - Totem-pole (push-pull)
 - Slew rate limit
 - Open drain
 - Driver Strength
- Configurable input buffer and pull settings:
 - Internal pull-up or pull-down options
 - Input sampling criteria
 - Input buffer can be disabled if not needed for lower power consumption
- Input event:
 - Up to four input event pins for each PORT group
 - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
 - Can be output to pin
- Selectable secured or non-secured attribution for each individual pin (only for PIC32CM SG devices)

33.3 Block Diagram

Figure 33-1. PORT I/O Block Diagram



33.4 Signal Description

Table 33-1. Signal Description for PORT

Signal name	Type	Description
Pxn	Digital I/O	General-purpose I/O, Port x, Pin n (Example: PA1, PC20)

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

33.5 I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device.

The following naming scheme is used: Each line bundle with up to 32 lines is assigned an identifier PORTx, with letter x = A, B, C... Each pin of PORTx is identified by a number n = 0, 1, ...31 for the nth pin of that Port. In this way PORT pins are identified as 'Pxn' (for example: PA24, PC3).

Each pin may be controlled by one or more peripheral multiplexer settings, which allow the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral has control over the output state of the pad, as well as the ability to read the current physical pad state. Refer to the [Pinout](#) for details.

On PIC32CM SG devices, each pin may be secured or nonsecured, with secured pins only accessible by secure accesses.

33.6 Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the Peripheral Clock Masking section of the Main Clock (MCLK).

The PORT requires an APB clock, which may be divided from the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

33.7 Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled. The PORT peripheral will continue operating in any sleep mode where its source clock is running.

33.8 Debug Operation

When the CPU is halted in debug mode, this peripheral will continue normal operation.

33.9 Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

Note: Optional write-protection is indicated by the "PAC Write-Protection" property in the register description. Write-protection does not apply for accesses through an external debugger.

33.10 Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

33.11 PIC32CM SG Specific Secure Access Protection

Depending on the H2PB configuration defined by the user in BOOTCFG fuses, the PORT module can either be configured as Secure or Mix-Secure. Secure or Non-secure transactions can then be filtered by the H2PB bridge. When a write is allowed by the H2PB bridge, it is still subject to the PAC write protection if enabled.

When the PORT is configured as Secure:

- Secure accesses are granted by the H2PB bridge
- Non-secure reads are blocked in the H2PB bridge, return 0 and generate a bus error
- Non-secure writes are blocked in the H2PB bridge, and generate a bus error

When the PORT is configured as Mix-secure, it is possible to open up individual pins for non-secure access. This is done using the NONSEC registers. When a pin has been set as non-secure, it can be handled from non-secure code. Since only Secured code has the rights to modify the NONSEC register, an interrupt-based mechanism has been added to let Non Secured code know when this register has been changed by Secured code. A single flag called NSCHK in the INTFLAG register will rise should changes, conditioned by the NSCHK register, occur in the NONSEC register.

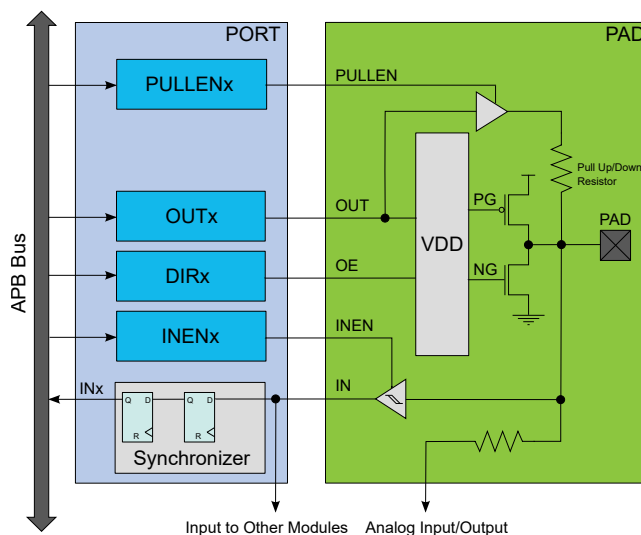
When the PORT is configured as Mix-secure, secure/non-secure accesses respect the following rules:

Table 33-2. PORT Register Access when Mix-Secure

ACCESS to	Access type	PORT Function
Non-Secure Registers : NSCHK, INTENCLR, INTENSET, INTFLAG	Secure access	Access granted
	Non-secure access	
Secure Registers: EVCTRL, CTRL	Secure access	Access granted
	Non-secure access	Read 0, Write Blocked, No bus error generated
Write-Secure Register NONSEC	Secure access	Access granted
	Non-secure access	Read Enabled, Write Blocked, No bus error generated
Mix-Secure Registers: DIR*, OUT*, WRCONFIG, PMUXn, PINCFGy, IN	Secure access	Access granted
	Non-secure access to a non-secure bit (NONSECx=1)	Access granted (per bit/register)
	Non-secure access to a secure bit (NONSECx=0)	Read 0, Write Blocked, No bus error generated (per bit/register)

33.12 Functional Description

Figure 33-2. Overview of the PORT

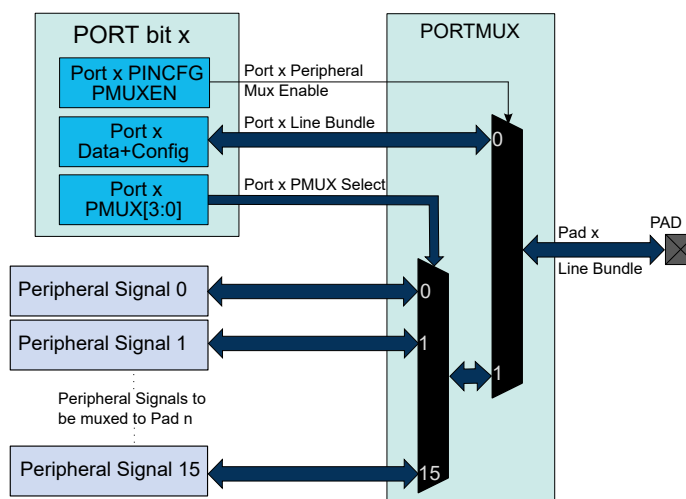


33.12.1 Principle of Operation

➔ Important: When $AVDD < 2.5V$, for proper operation of the analog functions on the I/O port pins it is critical that the internal charge pumps be enabled as necessary, as defined in `SUPC.VREGCTRL.CPEN[2:0]`.

Each PORT group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each PORT group, with increasing base addresses. The number of PORT groups may depend on the package/number of pins.

Figure 33-3. Overview of the Peripheral Functions Multiplexing



The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit "n" in OUT is written to '1', pin "n" is driven HIGH. If bit "n" in OUTx is written to '0', pin "n" is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGn) registers, with n=00, 01, ..31 representing the bit position.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers can be clocked only when system requires reading the input value, as specified in the SAMPLING field of the Control register (CTRL). The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGn.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGn register (PINCFGn.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing m (PMUXm, m=0,...15) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, connecting the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

On PIC32CM SG devices where the PORT peripheral is configured as mix-secure in H2PB, the security attribution of each pin in a PORT group is configured by the NONSEC register. If a bit in the NONSEC register is set to '0', the corresponding pin is configured as a secured pin and can only be handled by secure accesses. If a bit in the NONSEC register is set to '1', the corresponding pin is configured as a non-secured pin. Only secure accesses are allowed to write to the NONSEC register.

33.12.2 Basic Operation

33.12.2.1 Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled (Hi-Z tristate condition), even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.

33.12.2.2 Operation

Each I/O pin Pxn can be controlled by the registers in PORT. Each PORT group x has its own set of PORT registers, with a base address at byte address (PORT + 0x80 * group index) (A corresponds to group index 0, B to 1, and so on). Within that set of registers, the pin index is y, from 0 to 31.

Refer to the Pinout for additional information on available pin configuration and PORT groups.

Configuring Pins as Output

To use pin Pxn as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value, this must be done first before writing the DIR bit to avoid glitching the output pin prior to setting the user desired OUT value, so the previous value of the OUT bit is not driven on the pin.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

Configuring Pins as Input

To use pin Pxn as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in

that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGn.INEN) is written to '1'.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two cycles of the PORT clock. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

Using Alternative Peripheral Functions

To use pin Pxn as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGn register must be '1'. The PINCFGn register for pin Pxn is at byte offset (PINCFG0 + n).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXm register. The PMUXO/ PMUXE is at byte offset PMUX0 + (n/2). The chosen peripheral must also be configured and enabled.

33.12.3 I/O Pin Configurations

The Pin Configuration register (PINCFGn) is used for additional I/O pin configuration. A pin can be set in a totem-pole, open-drain, or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter and summarized in the following table:

Table 33-3. Pin Configurations Summary

DIR	INEN	PULLEN	OUT	Configuration
0	0	0	X	Reset or analog I/O: all digital disabled
0	0	1	0	Pull-down; input disabled
0	0	1	1	Pull-up; input disabled
0	1	0	X	Input
0	1	1	0	Input with pull-down
0	1	1	1	Input with pull-up
1	0	X	X	Output; input disabled
1	1	X	X	Output; input enabled

Note: X=Don't care.

33.12.3.1 Input Configuration

Figure 33-4. I/O Configuration - Standard Input

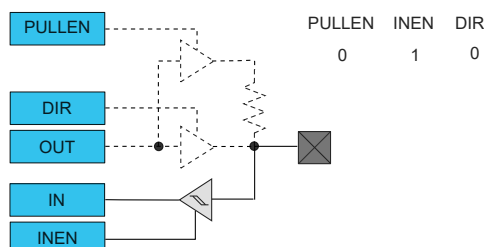
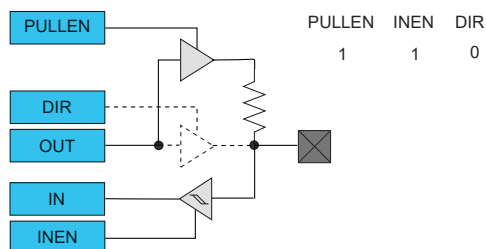


Figure 33-5. I/O Configuration - Input with Pull



Note: When pull is enabled, the pull value is defined by the OUT value.

33.12.3.2 Totem Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

Note: Enabling the output driver will automatically disable pull.

Figure 33-6. I/O Configuration - Totem-Pole Output with Disabled Input

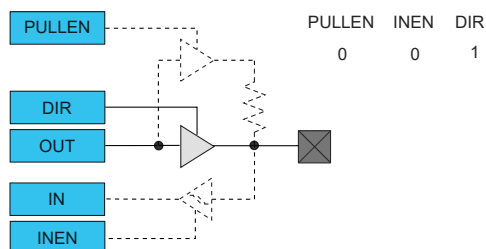


Figure 33-7. I/O Configuration - Totem-Pole Output with Enabled Input

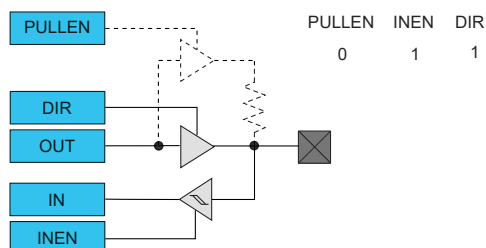
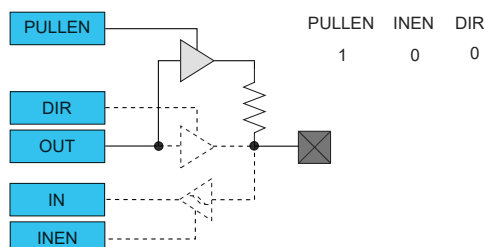


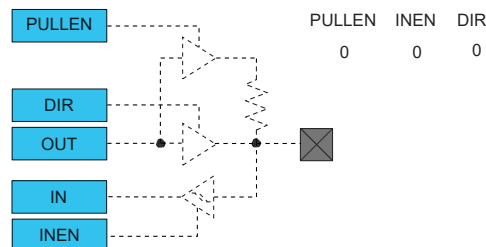
Figure 33-8. I/O Configuration - Output with Pull



33.12.3.3 Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

Figure 33-9. I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled



33.12.3.4 Secure Access Rights (Only on Devices with Security Attribution)

Non-secure write to CTRL, EVCTRL, or NONSEC registers is prohibited.

Non-secure read to CTRL or EVCTRL registers will return zero with no error resulting.

Non-secure write to a bit of DIR, DIRCLR, DIRSET, DIRTGL, OUT, OUTCLR, OUTSET, or OUTTGL registers is prohibited if the corresponding bit in NONSEC is zero.

Non-secure write to a PINCFGn register or to a PMUXn register field, either directly or through a write to the WRCONFIG register, is prohibited if the corresponding bit in NONSEC is zero.

DIR, DIRCLR, DIRSET, DIRTGL, IN, OUT, OUTCLR, OUTSET, or OUTTGL bits, PINCFGn registers, or PMUXn register fields relating to secure I/O pins (i.e. the corresponding bits in NONSEC are zero), read as zero in nonsecure mode, with no error resulting.

INTFLAG.NSCHK is set to 1 when NSCHK and NONSEC register values are different. Writing a 1 to INTFLAG.NSCHK will clear it and clear the PORT interrupt if enabled.

Secure code should initially write a 1 for all non-secure pins into both NONSEC and NSCHK registers. Then, whenever secure code writes a different value into NONSEC, INTFLAG.NSCHK will be set to 1 and a PORT interrupt will occur, if enabled. The non-secure code can then compare the values of the NONSEC and NSCHK registers to determine which PORT pins have just changed to secure or to non-secure. It should then copy the current NONSEC register value into NSCHK.

33.12.3.5 Events

The PORT allows input events to control individual I/O pins. These input events are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): I/O pin will be set when the incoming event has a high level ('1') and cleared when the incoming event has a low-level ('0').
- Set (SET): I/O pin will be set when an incoming event is detected.
- Clear (CLR): I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): I/O pin will toggle when an incoming event is detected.

The event is output to pin without any internal latency. For SET, CLEAR, and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.

The event actions can be configured with the Event Action m bit group in the Event Input Control register(EVCTRL.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) enables the corresponding action on input event. Writing '0' to this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to the *Event System (EVSYS)*, for details on configuring the Event System.

Each event input can address one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value

of the register (OUT) of this particular I/O pin, only one action is performed according to the table below.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

Table 33-4. Priority on Simultaneous SET/CLR/TGL Event Actions

EVACT0	EVACT1	EVACT2	EVACT3	Executed Event Action
SET	SET	SET	SET	SET
CLR	CLR	CLR	CLR	CLR
All Other Combinations				TGL

Be careful when the event is output to pin. Due to the fact the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

33.13 Peripheral Dependencies

Table 33-5. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
PORT	0x44800000	41 : NSCHK0/ NSCHK1/ NSCHK2/ NSCHK3	APB: CLKMSK[0] MASK27	---	20 STATUS0 PERID20		<u>Users:</u> EV0: USER[2],Resynch EV1: USER[3],Resynch EV2: USER[4],Resynch EV3: USER[5],Resynch	PD_CORE_SW

33.14 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	DIR	31:24					DIR[31:24]			
		23:16					DIR[23:16]			
		15:8					DIR[15:8]			
		7:0					DIR[7:0]			
0x04	DIRCLR	31:24					DIRCLR[31:24]			
		23:16					DIRCLR[23:16]			
		15:8					DIRCLR[15:8]			
		7:0					DIRCLR[7:0]			
0x08	DIRSET	31:24					DIRSET[31:24]			
		23:16					DIRSET[23:16]			
		15:8					DIRSET[15:8]			
		7:0					DIRSET[7:0]			
0x0C	DIRTGL	31:24					DIRTGL[31:24]			
		23:16					DIRTGL[23:16]			
		15:8					DIRTGL[15:8]			
		7:0					DIRTGL[7:0]			
0x10	OUT	31:24					OUT[31:24]			
		23:16					OUT[23:16]			
		15:8					OUT[15:8]			
		7:0					OUT[7:0]			
0x14	OUTCLR	31:24					OUTCLR[31:24]			
		23:16					OUTCLR[23:16]			
		15:8					OUTCLR[15:8]			
		7:0					OUTCLR[7:0]			
0x18	OUTSET	31:24					OUTSET[31:24]			
		23:16					OUTSET[23:16]			
		15:8					OUTSET[15:8]			
		7:0					OUTSET[7:0]			
0x1C	OUTTGL	31:24					OUTTGL[31:24]			
		23:16					OUTTGL[23:16]			
		15:8					OUTTGL[15:8]			
		7:0					OUTTGL[7:0]			
0x20	IN	31:24					IN[31:24]			
		23:16					IN[23:16]			
		15:8					IN[15:8]			
		7:0					IN[7:0]			
0x24	CTRL	31:24					SAMPLING[31:24]			
		23:16					SAMPLING[23:16]			
		15:8					SAMPLING[15:8]			
		7:0					SAMPLING[7:0]			
0x28	WRCONFIG	31:24	HWSEL	WRPINC	CFG	WRPMUX		PMUX[3:0]		
		23:16	DRVSTR[1:0]		SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
		15:8	PINMASK[15:8]							
		7:0	PINMASK[7:0]							
0x2C	EVCTRL	31:24	PORTEI3	EVACT3[1:0]		PID3[4:0]				
		23:16	PORTEI2	EVACT2[1:0]		PID2[4:0]				
		15:8	PORTEI1	EVACT1[1:0]		PID1[4:0]				
		7:0	PORTEI0	EVACT0[1:0]		PID0[4:0]				
0x30	PMUX0	7:0	PMUXO[3:0]				PMUXE[3:0]			
...										
0x3F	PMUX15	7:0	PMUXO[3:0]				PMUXE[3:0]			
0x40	PINCFG0	7:0	DRVSTR[1:0]		SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
...										
0x5F	PINCFG31	7:0	DRVSTR[1:0]		SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
0x60	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								NSCHK

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x64	INTENSET	31:24								
		23:16								
		15:8								
		7:0								NSCHK
0x68	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								NSCHK
0x6C	NONSEC	31:24	NONSEC[31:24]							
		23:16	NONSEC[23:16]							
		15:8	NONSEC[15:8]							
		7:0	NONSEC[7:0]							
0x70	NSCHK	31:24	NSCHK[31:24]							
		23:16	NSCHK[23:16]							
		15:8	NSCHK[15:8]							
		7:0	NSCHK[7:0]							

33.14.1 Data Direction

Name: DIR
Offset: 0x00
Reset: 0x00000000
Property: RW

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET), and Data Direction Clear (DIRCLR) registers.

Table 33-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DIR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DIR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DIR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIR[31:0] Port Data Direction

33.14.2 Data Direction Clear

Name: DIRCLR
Offset: 0x04
Reset: 0x00000000
Property: RW

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL), and Data Direction Set (DIRSET) registers.

Table 33-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIRCLR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DIRCLR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DIRCLR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DIRCLR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRCLR[31:0] Port Data Direction Clear

Writing a '0' to a bit has no effect.

Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as input.

33.14.3 Data Direction Set

Name: DIRSET
Offset: 0x08
Reset: 0x00000000
Property: RW

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL), and Data Direction Clear (DIRCLR) registers.

Table 33-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIRSET[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DIRSET[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DIRSET[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DIRSET[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRSET[31:0] Port Data Direction Set

Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin in the PORT group is configured as an output.

33.14.4 Data Direction Toggle

Name: DIRTGL
Offset: 0x0C
Reset: 0x00000000
Property: RW

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET), and Data Direction Clear (DIRCLR) registers.

Table 33-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DIRTGL[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DIRTGL[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DIRTGL[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DIRTGL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DIRTGL[31:0] Port Data Direction Toggle

Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The direction of the corresponding I/O pin is toggled.

33.14.5 Data Output Value

Name: OUT
Offset: 0x10
Reset: 0x00000000
Property: RW

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET), and Data Output Value Clear (OUTCLR) registers.

Table 33-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUT[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	OUT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	OUT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OUT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUT[31:0] PORT Data Output Value

33.14.6 Data Output Value Clear

Name: OUTCLR
Offset: 0x14
Reset: 0x00000000
Property: RW

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL), and Data Output Value Set (OUTSET) registers.

Table 33-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTCLR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	OUTCLR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	OUTCLR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OUTCLR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTCLR[31:0] PORT Data Output Value Clear

Writing '0' to a bit has no effect.

Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set the input pull direction to an internal pull-down.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down.

33.14.7 Data Output Value Set

Name: OUTSET
Offset: 0x18
Reset: 0x00000000
Property: RW

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL), and Data Output Value Clear (OUTCLR) registers.

Table 33-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTSET[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	OUTSET[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	OUTSET[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OUTSET[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTSET[31:0] PORT Data Output Value Set

Writing '0' to a bit has no effect.

Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

Value	Description
0	The corresponding I/O pin in the group will keep its configuration.
1	The corresponding I/O pin output is driven high or the input is connected to an internal pull-up.

33.14.8 Data Output Value Toggle

Name: OUTTGL
Offset: 0x1C
Reset: 0x00000000
Property: RW

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET), and Data Output Value Clear (OUTCLR) registers.

Table 33-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OUTTGL[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	OUTTGL[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	OUTTGL[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OUTTGL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – OUTTGL[31:0] PORT Data Output Value Toggle

Writing '0' to a bit has no effect.

Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

Value	Description
0	The corresponding I/O pin in the PORT group will keep its configuration.
1	The corresponding OUT bit value is toggled.

33.14.9 Data Input Value

Name: IN
Offset: 0x20
Reset: 0x00000000
Property: R

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Table 33-14. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN Value	Port WRCONFIG.PMUX Value	Peripheral Function	Description
0	N/A	Port	Normal Port in /out functions (Pxy)
1	00	EIC / EIC_EXTINT(n)	External Interrupts
1	01	ADC / CMP	ADC and Analog Comparator
1	03	SERCOM(n)	SERCOMn (UART, I ² C, SPI)
1	04	EBI	External Bus Interface
1	05	TCC WO(n)	Timer/Counter Controller
1	06	MLB	Media Local Bus
1	07	CAN(n) / SCL(n)	CAN, Serial Quad Interface
1	08	SDMMC	SD/MMC Host Controller (Memory Card Interface)
1	09	I2S / SWCLK, SWDIO, SWO, TRACE_CLK, TRACE_DATA[3:0]	I ² S audio and, Debug and debug Trace
1	0A	ETH	Ethernet
1	0B	Only 2 alternate ETH signals	RX_CLK on PD12 and GTX_CLK on PD05
1	0C	GCLK	Controls GCLK_IO[7:2]
1	0F	PTC	Peripheral Touch Controller

Note: All undefined values of WRCONFIG.PMUX are reserved.

Table 33-15. Port Pin Configuration Register Mapping

Port Group	Packages				
	100 pin TQFP EP	144 pin TQFP EP	144 pin TFBGA	176 pin TFBGA	208 pin TFBGA
PORT A (GROUP 0)	PA[0:20]	PA[0:29]	PA[0:29]	PA[0:31]	PA[0:31]
PINCFGn Registers	n=[0-20]	n=[0-29]	n=[0-29]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT B (GROUP 1)	PB[0:17]	PB[0:26]	PB[0:26]	PB[0:31]	PB[0:31]
PINCFGn Registers	n=[0-17]	n=[0-26]	n=[0-26]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT C (GROUP 2)	PC[0:15]	PC[0:29]	PC[0:29]	PC[0:31]	PC[0:31]
PINCFGn Registers	n=[0-15]	n=[0-29]	n=[0-29]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT D (GROUP 3)	PD[0:12]	PD[0:21]	PD[0:21]	PD[0:23]	PD[0:29]
PINCFGn Registers	n=[0-15]	n=[0-29]	n=[0-29]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT E (GROUP 4)	----	----	----	PE[0:3]	PE[0:11]

Table 33-15. Port Pin Configuration Register Mapping (continued)

Port Group	Packages				
	100 pin TQFP EP	144 pin TQFP EP	144 pin TFBGA	176 pin TFBGA	208 pin TFBGA
PINCFGn Registers	----	----	----	n=[0-3]	n=[0-11]
PINCFGn.SLEWLIM (Only Exists On:)	----	----	----	n=(0-3)	n=(0-4)
PORT F (GROUP 5)	----	----	----	PF[5]	PF[0:8]
PINCFGn Registers	----	----	----	n=[5]	n=[0-8]
PINCFGn.SLEWLIM (Only Exists On:)	----	----	----	----	n=(0-4)
PORT G (GROUP 6)	----	----	----	PG[0-3]	PG[0-11]
PINCFGn Registers	----	----	----	n=[0-3]	n=[0-11]
PINCFGn.SLEWLIM (Only Exists On:)	----	----	----	n=(0-3)	n=(0-4)

EXAMPLE:

PORT_REGS->GROUP[2].PORT_PINCFG4 = 0x12; /* I/O pin PC4, Slew rate control enabled (4x slower), input buffer enabled */

Table 33-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	IN[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	IN[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	IN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	IN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – IN[31:0] PORT Data Input Value

33.14.10 Control

Name: CTRL
Offset: 0x24
Reset: 0x00000000
Property: RW

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

Table 33-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SAMPLING[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SAMPLING[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SAMPLING[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SAMPLING[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SAMPLING[31:0] Input Sampling Mode

33.14.11 Write Configuration

Name: WRCONFIG
Offset: 0x28
Reset: 0x00000000
Property: W

This Write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

Table 33-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	HWSEL	WRPINCFC		WRPMUX	PMUX[3:0]			
Access	W	W		W	W	W	W	W
Reset	0	0		0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DRVSTR[1:0]		SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PINMASK[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PINMASK[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit 31 – HWSEL Half-Word Select

This bit selects the half-word field of a 32-PORT group to be reconfigured in the atomic write operation.

This bit will always read as zero.

Value	Description
0	The lower 16 pins of the PORT group will be configured.
1	The upper 16 pins of the PORT group will be configured.

Bit 30 – WRPINCFC Write PINCFG

This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.SLEWLIM, WRCONFIG.ODRAIN, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.

This bit will always read as zero.

Value	Description
0	The PINCFGn registers of the selected pins will not be updated.
1	The PINCFGn registers of the selected pins will be updated.

Bit 28 – WRPMUX Write PMUX

This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXm, m=0,...15) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.

Writing '0' to this bit has no effect.

Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG.PMUX value.

This bit will always read as zero.

Note: Not all PINCFGn registers are supported by all I/O port groups.

Value	Description
0	The PMUXm registers of the selected pins will not be updated.
1	The PMUXm registers of the selected pins will be updated.

Bits 27:24 – PMUX[3:0] Peripheral Multiplexing

These bits determine the new value written to the Peripheral Multiplexing register (PMUXm, m = 0,...15) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.

These bits will always read as zero.

Bits 23:22 – DRVSTR[1:0] Output Driver Strength Selection

Bits 21:20 – SLEWLIM[1:0] Output Driver Slew Rate Selection

This bit determines the new value written to PINCFGn.SLEWLIM for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Notes:

1. Only some pins have output TRISE/TFALL slew limit control.
2. Slew rate control can be used to improve signal integrity for high-speed signals if improper external HDW resistor termination was not utilized. One side effect however can be that if excessive slew is used it can affect maximum signal rates.
3. If an I²C function is enabled on a pin, set PINCFGn.SLEWLIM = 0x00 (disabled).

Value	Description
0x00	Slew rate control disabled (fast rise/fall time operation)
0x01	Slew rate control enabled (4x slower)
0x02	Slew rate control enabled (8x slower)
0x03	Slew rate control enabled (12x slower)

Bit 19 – ODRAIN Open Drain Output

This bit determines the new value written to PINCFGn.ODRAIN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.

This bit will always read as zero.

Value	Description
0	Output pin is Totem Pole (i.e. Push-Pull) output
1	Open drain output is enabled

Bit 18 – PULLEN Pull Enable

This bit determines the new value written to PINCFGn.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set. This bit will always read as zero.

Value	Description
0	Internal pin Pull-Up is disabled
1	Internal pin Pull-Up is enabled

Bit 17 – INEN Input Enable

This bit determines the new value written to PINCFGn.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set. This bit will always read as zero.

Value	Description
0	Disable input pin function
1	Enable input pin function

Bit 16 – PMUXEN Peripheral Multiplexer Enable

This bit determines the new value written to PINCFGn.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set. This bit will always read as zero.

Value	Description
0	The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled and the selected peripheral function controls the direction and output drive value.

Bits 15:0 – PINMASK[15:0] Pin Mask for Multiple Pin Configuration

These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit. These bits will always read as zero.

Value	Description
0	The configuration of the corresponding I/O pin in the half-word group will be left unchanged.
1	The configuration of the corresponding I/O pin in the half-word PORT group will be updated.

33.14.12 Event Input Control

Name: EVCTRL
Offset: 0x2C
Reset: 0x00000000
Property: RW

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

Table 33-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PORTEI3	EVACT3[1:0]		PID3[4:0]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PORTEI2	EVACT2[1:0]		PID2[4:0]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PORTEI1	EVACT1[1:0]		PID1[4:0]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PORTEI0	EVACT0[1:0]		PID0[4:0]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – PORTEI3 PORT Event Input Enable 3

Bits 30:29 – EVACT3[1:0] PORT Event Action 3

Bits 28:24 – PID3[4:0] PORT Event Pin Identifier 3

Bit 23 – PORTEI2 PORT Event Input Enable 2

Bits 22:21 – EVACT2[1:0] PORT Event Action 2

Bits 20:16 – PID2[4:0] PORT Event Pin Identifier 2

Bit 15 – PORTEI1 PORT Event Input Enable 1

Bits 14:13 – EVACT1[1:0] PORT Event Action 1

Bits 12:8 – PID1[4:0] PORT Event Pin Identifier 1

Bit 7 – PORTEI0 PORT Event Input Enable 0

Bits 6:5 – EVACT0[1:0] PORT Event Action 0

Value	Name	Description
0x0	OUT	Event output to pin
0x1	SET	Set output register of pin on event
0x2	CLR	Clear output register of pin on event
0x3	TGL	Toggle output register of pin on event

Bits 4:0 – PID0[4:0] PORT Event Pin Identifier 0

33.14.13 Peripheral Multiplexing

Name: PMUX
Offset: 0x30 + n*0x01 [n=0..15]
Reset: 0x00
Property: RW

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The m denotes the number of the set of I/O lines.

Table 33-20. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN Value	Port WRCONFIG.PMUX Value	Peripheral Function	Description
0	N/A	Port	Normal Port in /out functions (Pxy)
1	00	EIC / EIC_EXTINT(n)	External Interrupts
1	01	ADC / CMP	ADC and Analog Comparator
1	03	SERCOM(n)	SERCOMn (UART, I ² C, SPI)
1	04	EBI	External Bus Interface
1	05	TCC WO(n)	Timer/Counter Controller
1	06	MLB	Media Local Bus
1	07	CAN(n) / SQI(n)	CAN, Serial Quad Interface
1	08	SDMMC	SD/MMC Host Controller (Memory Card Interface)
1	09	I ² S / SWCLK, SWDIO, SWO, TRACE_CLK, TRACE_DATA[3:0]	I ² S audio and, Debug and debug Trace
1	0A	ETH	Ethernet
1	0B	Only 2 alternate ETH signals	RX_CLK on PD12 and GTX_CLK on PD05
1	0C	GCLK	Controls GCLK_IO[7:2]
1	0F	PTC	Peripheral Touch Controller

Note: All undefined values of WRCONFIG.PMUX are reserved.

Table 33-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	PMUXO[3:0]				PMUXE[3:0]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:4 – PMUXO[3:0] Peripheral Multiplexing for Odd-Numbered Pin

These bits select the peripheral function for odd-numbered pins ($2*n + 1$) of a PORT group, if the corresponding PINCFGn.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the Pinout section.

Table 33-23. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN Value	Port WRCONFIG.PMUX Value	Peripheral Feature	Peripheral Function	Description
0	N/A	Port	----	Normal Port in /out functions (Pxy)
1	0x00	EIC / EIC_EXTINT(n)	A	External Interrupts
1	0x01	ADC / CMP	B	ADC and Analog Comparator
1	0x03	SERCOM(n)	D	SERCOMn (UART, I ² C, SPI)
1	0x04	EBI	E	External Bus Interface
1	0x05	TCC[9:0] WO(n)	F	Timer/Counter Controller
1	0x02	TCC[2] WO(n)	C	
1	0x06	MLB	G	Media Local Bus
1	0x07	CAN(n) / SQR(n)	H	CAN, Serial Quad Interface
1	0x08	SDMMC	I	SD/MMC Host Controller (Memory Card Interface)
1	0x09	I ² S / SWCLK, SWDIO, SWO, TRACE_CLK, TRACE_DATA[3:0]	J	I ² S audio and, Debug and debug Trace
1	0x0A	ETH / GMAC	K	Ethernet
1	0x0B	Only 2 alternate ETH signals	L	RX_CLK on PD12 and GTX_CLK on PD05
1	0x0C	GCLK_IOn	M	Controls GCLK_IO[7:2]
---	0x0D	Reserved	---	Reserved
---	0x0E	Reserved	---	Reserved
1	0x0F	PTC	P	Peripheral Touch Controller

Value	Name	Description
0	A	Peripheral function A selected
1	B	Peripheral function B selected
3	D	Peripheral function D selected
5	F	Peripheral function F selected
7	H	Peripheral function H selected
8	I	Peripheral function I selected
10	K	Peripheral function K selected
15	P	Peripheral function P selected

Bits 3:0 – PMUXE[3:0] Peripheral Multiplexing for Even-Numbered Pin

These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGn.PMUXEN bit is '1'.

Not all possible values for this selection may be valid. For more details, refer to the Pinout.

Table 33-22. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN Value	Port WRCONFIG.PMUX Value	Peripheral Feature	Peripheral Function	Description
0	N/A	Port	----	Normal Port in /out functions (Pxy)
1	0x00	EIC / EIC_EXTINT(n)	A	External Interrupts
1	0x01	ADC / CMP	B	ADC and Analog Comparator
1	0x03	SERCOM(n)	D	SERCOMn (UART, I ² C, SPI)
1	0x04	EBI	E	External Bus Interface
1	0x05	TCC[9:0] WO(n)	F	Timer/Counter Controller
1	0x02	TCC[2] WO(n)	C	
1	0x06	MLB	G	Media Local Bus
1	0x07	CAN(n) / SQR(n)	H	CAN, Serial Quad Interface

Table 33-22. Peripheral Port Mux Control Mapping (continued)

Port PINCFGn.MUXEN Value	Port WRCONFIG.PMUX Value	Peripheral Feature	Peripheral Function	Description
1	0x08	SDMMC	I	SD/MMC Host Controller (Memory Card Interface)
1	0x09	I ² S / SWCLK, SWDIO, SWO, TRACE_CLK, TRACE_DATA[3:0]	J	I ² S audio and, Debug and debug Trace
1	0x0A	ETH / GMAC	K	Ethernet
1	0x0B	Only 2 alternate ETH signals	L	RX_CLK on PD12 and GTX_CLK on PD05
1	0x0C	GCLK_IOn	M	Controls GCLK_IO[7:2]
---	0x0D	Reserved	---	Reserved
---	0x0E	Reserved	---	Reserved
1	0x0F	PTC	P	Peripheral Touch Controller

Value	Name	Description
0	A	Peripheral function A selected
1	B	Peripheral function B selected
3	D	Peripheral function D selected
5	F	Peripheral function F selected
7	H	Peripheral function H selected
8	I	Peripheral function I selected
10	K	Peripheral function K selected
15	P	Peripheral function P selected

33.14.14 Pin Configuration

Name: PINCFG
Offset: 0x40 + n*0x01 [n=0..31]
Reset: 0x00
Property: RW

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

Table 33-24. Peripheral Port Mux Control Mapping

Port PINCFGn.MUXEN Value	Port WRCONFIG.PMUX Value	Peripheral Function	Description
0	N/A	Port	Normal Port in /out functions (Pxy)
1	00	EIC / EIC_EXTINT(n)	External Interrupts
1	01	ADC / CMP	ADC and Analog Comparator
1	03	SERCOM(n)	SERCOMn (UART, I ² C, SPI)
1	04	EBI	External Bus Interface
1	05	TCC WO(n)	Timer/Counter Controller
1	06	MLB	Media Local Bus
1	07	CAN(n) / SQL(n)	CAN, Serial Quad Interface
1	08	SDMMC	SD/MMC Host Controller (Memory Card Interface)
1	09	I2S / SWCLK, SWDIO, SWO, TRACE_CLK, TRACE_DATA[3:0]	I ² S audio and, Debug and debug Trace
1	0A	ETH	Ethernet
1	0B	Only 2 alternate ETH signals	RX_CLK on PD12 and GTX_CLK on PD05
1	0C	GCLK	Controls GCLK_IO[7:2]
1	0F	PTC	Peripheral Touch Controller

Note: All undefined values of WRCONFIG.PMUX are reserved.

Table 33-25. Port Pin Configuration Register Mapping

Port Group	Packages				
	100 pin TQFP EP	144 pin TQFP EP	144 pin TFBGA	176 pin TFBGA	208 pin TFBGA
PORT A (GROUP 0)	PA[0:20]	PA[0:29]	PA[0:29]	PA[0:31]	PA[0:31]
PINCFGn Registers	n=[0-20]	n=[0-29]	n=[0-29]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT B (GROUP 1)	PB[0:17]	PB[0:26]	PB[0:26]	PB[0:31]	PB[0:31]
PINCFGn Registers	n=[0-17]	n=[0-26]	n=[0-26]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT C (GROUP 2)	PC[0:15]	PC[0:29]	PC[0:29]	PC[0:31]	PC[0:31]
PINCFGn Registers	n=[0-15]	n=[0-29]	n=[0-29]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT D (GROUP 3)	PD[0:12]	PD[0:21]	PD[0:21]	PD[0:23]	PD[0:29]
PINCFGn Registers	n=[0-15]	n=[0-29]	n=[0-29]	n=[0-31]	n=[0-31]
PINCFGn.SLEWLIM (Only Exists On:)	n=(0-4)	n=(0-4, 21)	n=(0-4, 21)	n=(0-4, 21, 30, 31)	n=(0-4, 21, 30, 31)
PORT E (GROUP 4)	----	----	----	PE[0:3]	PE[0:11]

Table 33-25. Port Pin Configuration Register Mapping (continued)

Port Group	Packages				
	100 pin TQFP EP	144 pin TQFP EP	144 pin TFBGA	176 pin TFBGA	208 pin TFBGA
PINCFGn Registers	----	----	----	n=[0-3]	n=[0-11]
PINCFGn.SLEWLIM (Only Exists On:)	----	----	----	n=(0-3)	n=(0-4)
PORT F (GROUP 5)	----	----	----	PF[5]	PF[0:8]
PINCFGn Registers	----	----	----	n=[5]	n=[0-8]
PINCFGn.SLEWLIM (Only Exists On:)	----	----	----	----	n=(0-4)
PORT G (GROUP 6)	----	----	----	PG[0-3]	PG[0-11]
PINCFGn Registers	----	----	----	n=[0-3]	n=[0-11]
PINCFGn.SLEWLIM (Only Exists On:)	----	----	----	n=(0-3)	n=(0-4)

EXAMPLE:

PORT_REGS->GROUP[2].PORT_PINCFG4 = 0x12; /* I/O pin PC4, Slew rate control enabled (4x slower), input buffer enabled */

Table 33-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	DRVSTR[1:0]		SLEWLIM[1:0]		ODRAIN	PULLEN	INEN	PMUXEN
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:6 – DRVSTR[1:0] Output Driver Strength Selection

Bits 5:4 – SLEWLIM[1:0] Output Driver Slew Rate Selection

Notes:

1. Only some pins have output TRISE/TFALL slew limit control. See Table: ____ Port Pin Configuration Register Mapping below for details.
2. Slew rate control can be used to improve signal integrity for high speed signals if improper external HDW resistor termination was not utilized.
3. If an I²C function is enabled on a pin, the corresponding PINCFGn.SLEWLIM MUST = 0x00.

Value	Name	Description
0x0	FAST	Slew rate control disabled (fast rise/fall time operation)
0x1	SLOW4	Slew rate control enabled (4x slower)
0x2	SLOW8	Slew rate control enabled (8x slower)
0x3	SLOW12	Slew rate control enabled (12x slower)

Bit 3 – ODRAIN Open Drain Output

Value	Description
0	The open drain output is disabled.
1	The open drain output is enabled.

Bit 2 – PULLEN Pull Enable

This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

Value	Description
0	Internal pull resistor is disabled and the input is in a high-impedance configuration.
1	Internal pull resistor is enabled and the input is driven to a defined logic level in the absence of external input.

Bit 1 – INEN Input Enable

This bit controls the input buffer of an I/O pin configured as either an input or output. Writing a zero to this bit disables the input buffer completely, preventing read-back of the Physical Pin state when the pin is configured as either an input or output.

Value	Description
0	Input buffer for the I/O pin is disabled and the input value will not be sampled.
1	Input buffer for the I/O pin is enabled and the input value will be sampled when required.

Bit 0 – PMUXEN Peripheral Multiplexer Enable

This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXm, m=0,...15) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.

Writing a zero to this bit allows the PORT to control the pad direction via the Data Direction register (DIR) and output logic level via the Data Output Value register (OUT). The peripheral multiplexer value in PMUXm is ignored. Writing '1' to this bit enables the peripheral selection in PMUXm to control the pad. In this configuration, the Physical Pin state may still be read from the Data Input Value register (IN) if PINCFGn.INEN is set.

Value	Description
0	The peripheral multiplexer selection is disabled and the PORT registers control the direction and output drive value.
1	The peripheral multiplexer selection is enabled and the selected peripheral function controls the direction and output drive value.

33.14.15 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x60
Reset: 0x00000000
Property: RW



Important: This register is only present in devices with security attribution.

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Table 33-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Non-Secure Check Interrupt Enable bit, which disables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

33.14.16 Interrupt Enable Set

Name: INTENSET
Offset: 0x64
Reset: 0x00000000
Property: RW



Important: This register is only present in devices with security attribution.

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Table 33-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Non-Secure Check Interrupt Enable bit, which enables the Non-Secure Check interrupt.

Value	Description
0	The Non-Secure Check interrupt is disabled.
1	The Non-Secure Check interrupt is enabled.

33.14.17 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x68
Reset: 0x00000000
Property: RW



Important: This register is only present in devices with security attribution.

Table 33-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								NSCHK
Access								RW
Reset								0

Bit 0 – NSCHK Non-Secure Check

This flag is set on NONSEC write when a bit in NSCHK is 1 and the corresponding bit in NONSEC is cleared, or when a bit in NSCHK is 0 and the corresponding bit in NONSEC is set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Secure Check interrupt flag.

33.14.18 Security Attribution

Name: NONSEC
Offset: 0x6C
Reset: 0x00000000
Property: RW



Important: This register is only present in devices with security attribution.

This register allows the user to configure one or more I/O pins as secured or non-secured.



Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Table 33-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NONSEC[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NONSEC[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NONSEC[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NONSEC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NONSEC[31:0] Port Security Attribution

These bits set the security attribution for the individual I/O pins in the PORT group.

Value	Description
0	The corresponding I/O pin in the PORT group is configured as secured. When module is PAC secured, the configuration for this pin is only available through the secure alias. Attempt to change the pin configuration through the non-secure alias will be silently ignored and reads will return 0.
1	The corresponding I/O pin in the PORT group is configured as non-secured. The I/O line configuration for this pin is available through the non-secure alias.

33.14.19 Security Attribution Check

Name: NSCHK
Offset: 0x70
Reset: 0x00000000
Property: RW



Important: This register is only present in devices with security attribution.

This register allows the user to select one or more pins to check their security attribution as non-secured.



Tip: The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

Table 33-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NSCHK[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	NSCHK[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NSCHK[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NSCHK[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NSCHK[31:0] Port Security Attribution Check

These bits select the individual pins for security attribution check. If any pin selected in NSCHK has the corresponding bit in NONSEC set to the opposite value, then the NSCHK interrupt flag will be set.

Value	Description
0	0-to-1 transition will be detected on corresponding NONSEC bit.
1	1-to-0 transition will be detected on corresponding NONSEC bit.

34. Serial Communication Interface (SERCOM)

34.1 Overview

There are up to six instances of the Serial Communication interface (SERCOM) peripheral.

Each SERCOM provides hardware support for one of the I²C, SPI, LIN, or USART protocols. It can operate as an I²C Host, I²C Client, SPI Host, SPI Client, LIN Host, LIN Client, or a USART. When a SERCOM is configured and enabled, the resources of that SERCOM are dedicated to the selected protocol and operational mode. It can use the internal generic clock or an external clock which allows it to operate in all sleep modes.

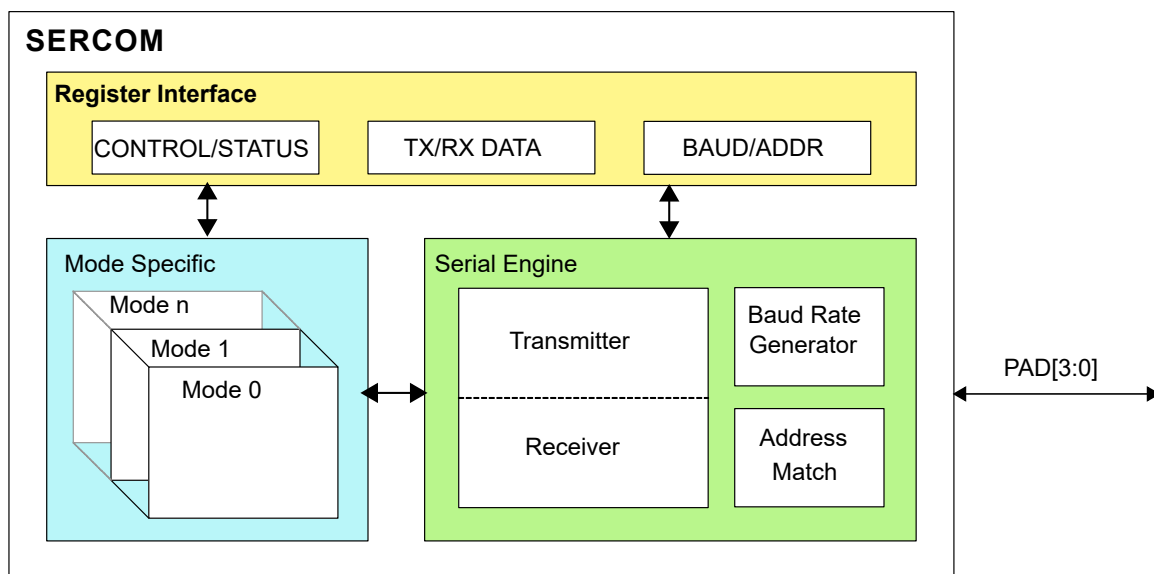
34.2 Features

The following are key features of the module:

- Interface for configuring operating mode:
 - Inter-Integrated Circuit (I²C) two-wire serial interface
 - System Management Bus (SMBus™) compatible
 - Serial Peripheral Interface (SPI)
 - Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
(Plus LIN Host, LIN Client when operating in USART mode)
- Baud Rate Generator (BRG)
- Address Match/mask Logic
- Operational in all Sleep modes with an External Clock Source
- Can be used with DMA
- Up to 16-byte Internal FIFOs for Transmit and Receive

34.3 Block Diagram

Figure 34-1. SERCOM Block Diagram



34.4 Functional Description

34.4.1 Principle of Operation

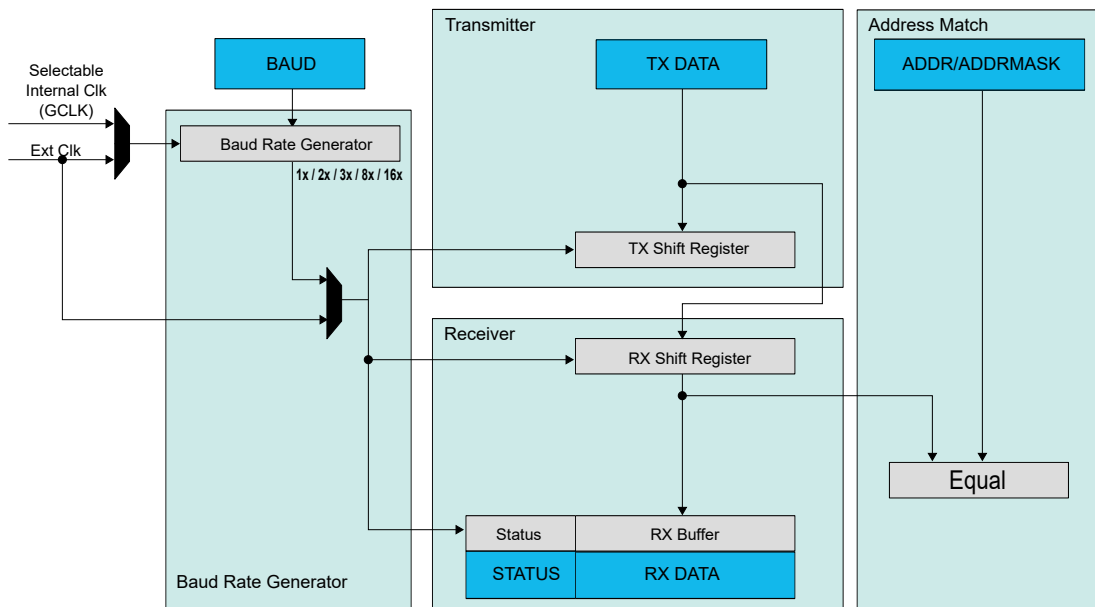
The basic structure of the SERCOM serial engine is shown in the following figure.

- The transmitter consists of a single write buffer and a Shift register
- The receiver consists of a one-level (I²C), two-level (USART, SPI) receive buffer and a Shift register

The Baud Rate Generator (BRG) is capable of running on the GCLK_SERCOMn_CORE clock or an external clock.

Address matching logic is included for SPI and I²C operation.

Figure 34-2. SERCOM Serial Engine



34.4.2 Basic Operation

34.4.2.1 Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE).

Table 34-1. SERCOM Modes

CTRLA.MODE	Description
0x0	USART with external clock
0x1	USART with internal clock
0x2	SPI in client operation
0x3	SPI in host operation
0x4	I ² C client operation
0x5	I ² C host operation
0x6-0x7	Reserved

34.4.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register descriptions (one for each mode of operation) for details.

- USART.CTRLA
- SPI.CTRLA
- I²C Host CTRLA
- I²C Client CTRLA

34.4.2.3 Clock Generation – Baud-Rate Generator

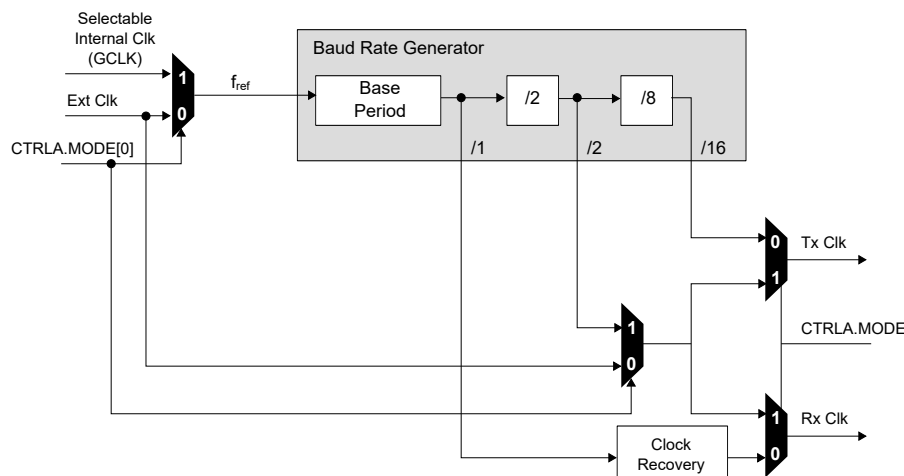
Each SERCOM contains a baud rate generator as shown in the following figure, which can generate an internal clock for asynchronous or synchronous communication. The output frequency (f_{BAUD}) is determined by the Baud register (BAUD) setting and the baud reference clock frequency (f_{ref}). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

Figure 34-3. Baud Rate Generator



The following table contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For Asynchronous Arithmetic mode, the BAUD register value is 16 bits (0 to 65,535).

For Asynchronous Fractional mode, the BAUD register value contains 13 bits of integer (BAUD.BAUD) as well as a 3 bit fractional adjustment (BAUD.FP). In this mode the BAUD setting must be greater than or equal to 1. This mode is used for auto baud detection. See CTRLA.FORM and CTRLA.SAMPR.

For Synchronous operation, the BAUD register value is 8 bits (0 to 255). CTRLA.CMODE=1.

Table 34-2. Baud Rate Equations

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Arithmetic	$f_{BAUD} \leq \frac{f_{ref}}{16}$	$f_{BAUD} = \frac{f_{ref}}{16} \left(1 - \frac{BAUD}{65536} \right)$	$BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}} \right)$

Table 34-2. Baud Rate Equations (continued)

Operating Mode	Condition	Baud Rate (Bits Per Second)	BAUD Register Value Calculation
Asynchronous Fractional	$f_{BAUD} \leq \frac{f_{ref}}{S}$	$f_{BAUD} = \frac{f_{ref}}{S \cdot (BAUD + \frac{FP}{8})}$	$BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$
Synchronous	$f_{BAUD} \leq \frac{f_{ref}}{2}$	$f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$	$BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$

S = Number of samples per bit, which can be 16, 8, or 3.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}} \right)$$

Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for f_{BAUD} calculates the average frequency over 65536 f_{ref} cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of f_{BAUD} over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

where:

- D represent the data bits per frame
- S represent the sum of start and first stop bits, if present.

The following table shows the BAUD register value versus baud frequency f_{BAUD} at a serial engine frequency of 48 MHz. This assumes a D value of 8 bits and an S value of 2 bits (10 bits, including start and stop bits).

Table 34-3. BAUD Register Value vs. Baud Frequency

BAUD Register Value	Serial Engine CPF	f_{BAUD} at 48MHz Serial Engine Frequency (f_{REF})
0 – 406	160	3 MHz
407 – 808	161	2.981 MHz
809 – 1205	162	2.963 MHz
...	...	...
65206	31775	15.11 kHz
65207	31871	15.06 kHz
65208	31969	15.01 kHz

34.4.3 Additional Features

34.4.3.1 Address Match and Mask

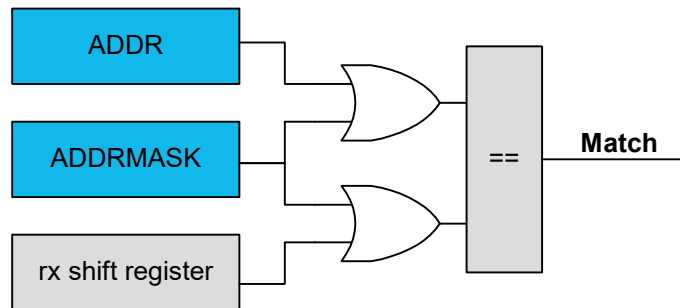
In either SPI client or I²C client mode, the SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

34.4.3.1.1 Address With Mask (CTRLB.AMODE = 0X0)

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match if the bit values in the shift register match the values in the Address register OR the associated bit in the Address Mask register is a 1. All bits that are masked are not included in the match. Note

that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

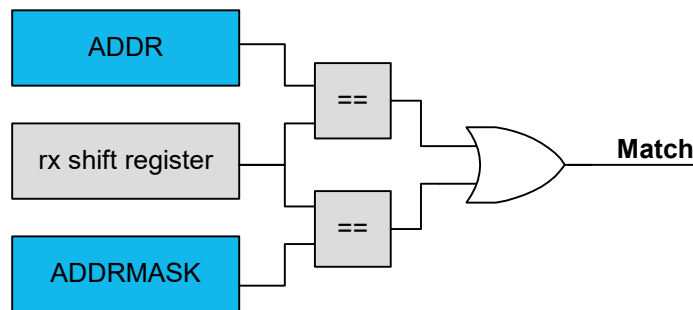
Figure 34-4. Address With Mask



34.4.3.1.2 Two Unique Addresses (CTRLB.AMODE = 0X1)

Either of the two addresses written to ADDR and ADDRMASK will cause a match.

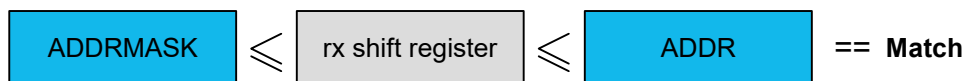
Figure 34-5. Two Unique Addresses



34.4.3.1.3 Address Range (CTRLB.AMODE = 0X2)

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

Figure 34-6. Address Range



34.4.3.1.4 Sleep Mode Operation

The peripheral can operate in any Sleep mode where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake-up the device from Sleep modes. Refer to the different SERCOM mode chapters for details.

34.4.3.1.5 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

34.5 Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART)

34.5.1 USART Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver, BAUD Rate Generator, and either an internal or external clock.

The transmitter consists of a 0-byte FIFO, a Shift register, and control logic for different frame formats. The write buffer supports data transmission without any delay between frames. The receiver consists of a 0-byte FIFO and a Shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

34.5.2 USART Features

- Full-duplex Operation
- Asynchronous (with Clock Reconstruction) or Synchronous Operation
- Internal or External Clock source for Asynchronous and Synchronous Operation
- Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data bits and 1 or 2 Stop bits
- Odd or Even Parity Generation and Parity Check
- Selectable LSB or MSB-first Data Transfer
- Buffer Overflow and Frame Error Detection
- Noise Filtering, Including False Start bit Detection and Digital Low-pass Filter
- Collision Detection
- Can Operate in all Sleep modes
- Operation at Speeds up to Half the System Clock for Internally Generated Clocks
- Operation at Speeds up to the System Clock for Externally Generated Clocks
- RTS and CTS Flow Control
- IrDA Modulation and Demodulation up to 115.2 kbps
- LIN Host Support
- LIN Client Support
- Auto-baud and break character detection
- ISO 7816 T = 0 or T = 1 protocols for Smart Card Interfacing
- RS485 Support
- Start-of-frame detection
- 16-Byte Receive FIFO with FIFO enabled
- 16-Byte Transmit FIFO with FIFO enabled
- USART DMA Support
- 32-bit Extension for Better System Bus Utilization

34.5.3 USART Signal Descriptions

Table 34-4. SERCOM USART Signal Descriptions

Signal Names	Function	Pin Type	Description
SERCOMn_PAD[0] SERCOMn_PAD[1] SERCOMn_PAD[2] SERCOMn_PAD[3]	TXD	Digital Output	USART TX Transmit Data
	RXD	Digital Input	USART RX Receive Data
	XCK	Digital Input/Output	USART external clock input or output
	RTS	Digital Output	USART Flow Control “Request-To-Send” output
	CTS	Digital Input	USART Flow Control “Clear-To-Send” Input
	TE	Digital Output	USART “Transmit Enable” output in RS485 mode

Table 34-5. USART User Configuration Summary

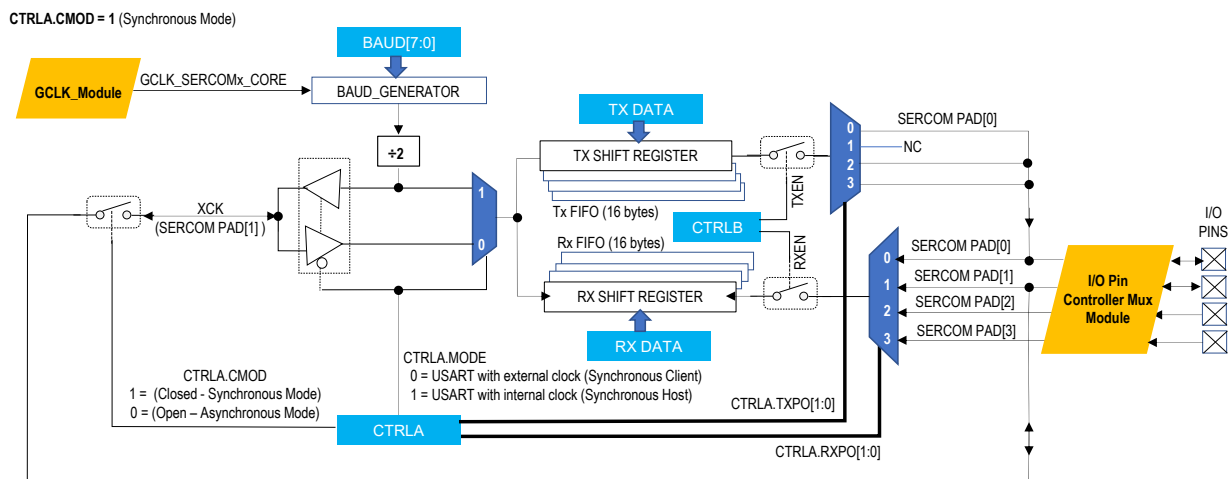
CMODE	FORM	TXPO	RXPO	SAMPR	ENC	LINCMD	PMODE	MODE	HOST	CLIENT	TXINV	RXINV	DORD	CHSIZE	TXEN/RXEN /ENABLE	SBMODE	GTIME	RTS/CTS/TE	COMMENTS														
1 (Sync)	0x0		0x2	—			—	0	NO	YES	X	X		0x0, 0x1, 0x5, 0x6, 0x7		0	X	—	(Client) FD Sync, Ext Clk in														
			0x3				—	1	YES	NO	X	X						—	(Host) FD Sync Ext Clk out														
							—	0	NO	YES	X	X						—	(Client) FD Sync Ext Clk in														
			0x1				0x2	—	1	YES	NO	X						X	—	(Host) FD Sync Ext Clk out													
	0x3	X						0	NO	YES	X	X						—	(Client) FD Sync Ext Clk in														
		1						YES	NO	X	X	—						(Host) FD Sync Ext Clk out															
	0x7	0x0						0	0	NO	YES							1	1	X	0x0	0	2	—	ISO 7816 (Client) HD Sync Ext Clk in								
			1				YES		NO			—						ISO 7816 (Host) HD Sync Ext Clk Out															
			0 (Async)				0x0		0x0	0x2,3	0x0,0x2,0x4	0						—	1					—	—	X	X	0x0, 0x1, 0x5, 0x6, 0x7	1/1/1	X	X	—	RS-232 FD without Flow Control
									0x2	0x1								—	1					—	—	X	X				RTS/CTS	RS-232 FD w/ Flow Control	
	0x3	0x3						—	1	—								—	X	X	1	TE	RS-485 (HD/FD, No Flow Ctrl)										
	0x1	0x0						0x2,3	X	1								—	—	X	X	X	—	RS-232 FD without Flow Control									
0x2		0x1		X	1	—		—	X	X				RTS/CTS	RS-232 FD w/ Flow Control																		
0x3	0x3	X		1	—	—		X	X	1				TE	RS-485 (HD/FD, No Flow Ctrl)																		
0x2	0x0	0x2,3		0x0,0x1	0x2												X	—	—	LIN Host – HDW Break, Sync, and ID generation													
					0x1	—	1	YES	NO	0				0	1				0	—	LIN Host – HDW Break gen, SW Sync and SW ID generation												
0x4	0x0	0x2,3					—	—	1	NO	YES	0	0	1	0		X	—	—	Auto-baud (LIN Client) - break detection (Fractional Baud)													
							—	X	1	NO	YES	0	0	1	0				—	Auto-baud (LIN Client) - break detection w/													
	0x5	0x0		0x2,3		0x1		—	X	1	NO	YES	0	0	1	0		X	—	Auto-baud (LIN Client) - break detection w/													

Table 34-5. USART User Configuration Summary (continued)

CMODE	FORM	TXPO	RXPO	SAMPR	ENC	LINCMD	PMODE	MODE	HOST	CLIENT	TXINV	RXINV	DORD	CHSIZE	TXEN/RXEN /ENABLE	SBMODE	GTIME	RTS/CTS/TE	COMMENTS
Legend X = Application specific, user selectable HD = Half Duplex FD = Full Duplex — = Not applicable																			

34.5.4 SERCOM USART Synchronous Mode

Figure 34-7. Simplified USART Synchronous Mode Block Diagram

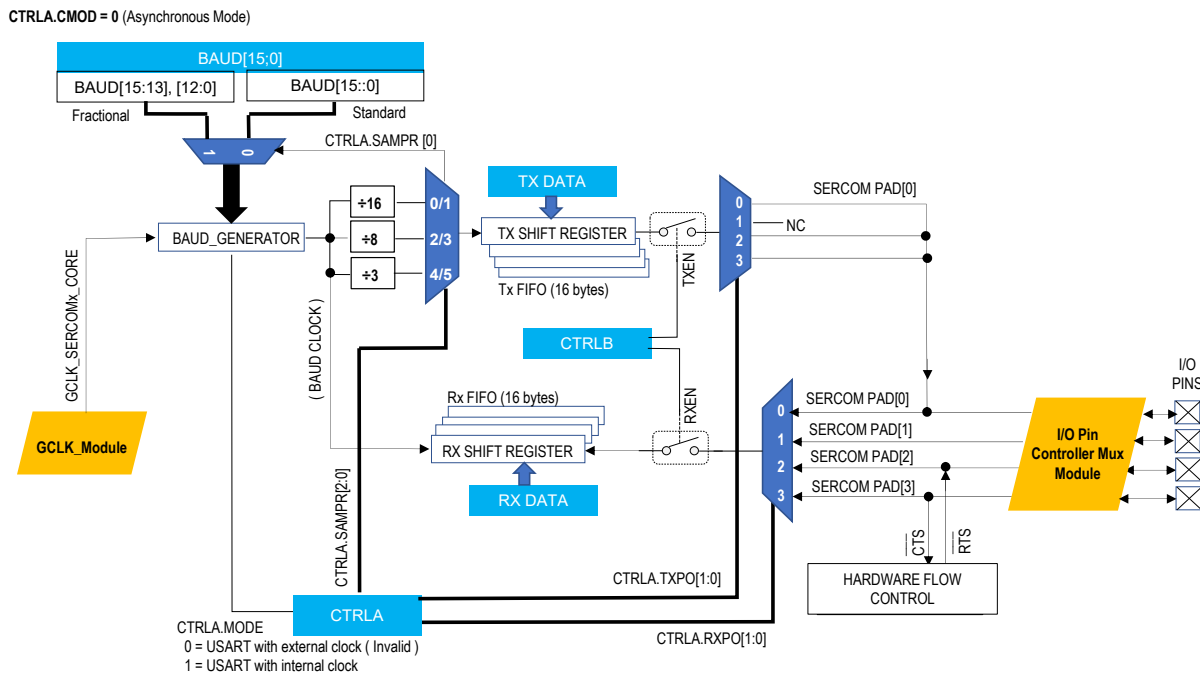


Notes:

1. In Synchronous USART Client Mode, USART - CTRLA.MODE = 0, USART RX/TX clock is supplied externally through input on SERCOM PAD[1], USART - CTRLA.TXPO = 0x0 or 0x3 from external remote USART device.
2. In Synchronous USART Host Mode, USART - CTRLA.MODE = 1, USART RX/TX shift register clock sources are supplied internally from GCLK_SERCOMx_CORE and output on SERCOM PAD[1] to external remote USART device.

34.5.5 SERCOM USART Asynchronous Mode Block Diagram

Figure 34-8. Simplified USART Asynchronous Mode Block Diagram



34.5.6 SERCOM USART Peripheral Dependencies

Table 34-6. SERCOM USART Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AXI/APB BUS Clocks	GCLK PCHCTRLn Peripheral Channel Index: Clock Name	PAC Peripheral Peripheral Identifier (PAC.WRCTRL)	DMA Trigger Index: Source	Power Domain
SERCOM0	0x4600_0000	55 : ERROR 56 : RXBRK 57 : DRE 58 : TXC 59 : RXC 60 : RXS 61 : CTSIC	CLKMSK0[31]	21 : GCLK_SERCOM0_CORE	WRCTRL.PERID[23]	CHCTRLBk.TRIG = 0x5 : RX CHCTRLBk.TRIG = 0x6 : TX	VDDREG
SERCOM1	0x4600_2000	62 : ERROR 63 : RXBRK 64 : DRE 65 : TXC 66 : RXC 67 : RXS 68 : CTSIC	CLKMSK1[0]	22 : GCLK_SERCOM1_CORE	WRCTRL.PERID[24]	CHCTRLBk.TRIG = 0x7 : RX CHCTRLBk.TRIG = 0x8 : TX	VDDREG

Table 34-6. SERCOM USART Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AXI/APB BUS Clocks	GCLK PCHCTRLn Peripheral Channel Index: Clock Name	PAC Peripheral Peripheral Identifier (PAC.WRCTRL)	DMA Trigger Index: Source	Power Domain
SERCOM2	0x4580_0000	69 : ERROR 70 : RXBRK 71 : DRE 72 : TXC 73 : RXC 74 : RXS 75 : CTSIC	CLKMSK1[1]	23 : GCLK_SERCOM2_CORE	WRCTRL.PERID[25]	CHCTRLBk.TRIG = 0x9 : RX CHCTRLBk.TRIG = 0x10 : TX	VDDREG
SERCOM3	0x4580_2000	76 : ERROR 77 : RXBRK 78 : DRE 79 : TXC 80 : RXC 81 : RXS 82 : CTSIC	CLKMSK1[2]	24 : GCLK_SERCOM3_CORE	WRCTRL.PERID[26]	CHCTRLBk.TRIG = 0x11 : RX CHCTRLBk.TRIG = 0x12 : TX	VDDREG
SERCOM4	0x4600_4000	83 : ERROR 84 : RXBRK 85 : DRE 86 : TXC 87 : RXC 88 : RXS 89 : CTSIC	CLKMSK1[3]	25 : GCLK_SERCOM4_CORE	WRCTRL.PERID[27]	CHCTRLBk.TRIG = 0x13 : RX CHCTRLBk.TRIG = 0x14 : TX	VDDREG
SERCOM5	0x4580_4000	90 : ERROR 91 : RXBRK 92 : DRE 93 : TXC 94 : RXC 95 : RXS 96 : CTSIC	CLKMSK1[4]	26 : GCLK_SERCOM5_CORE	WRCTRL.PERID[28]	CHCTRLBk.TRIG = 0x15 : RX CHCTRLBk.TRIG = 0x16 : TX	VDDREG
SERCOM6	0x4580_6000	97 : ERROR 98 : RXBRK 99 : DRE 100 : TXC 101 : RXC 102 : RXS 103 : CTSIC	CLKMSK1[5]	27 : GCLK_SERCOM6_CORE	WRCTRL.PERID[29]	CHCTRLBk.TRIG = 0x17 : RX CHCTRLBk.TRIG = 0x18 : TX	VDDREG
SERCOM7	0x4500_0000	104 : ERROR 105 : RXBRK 106 : DRE 107 : TXC 108 : RXC 109 : RXS 110 : CTSIC	CLKMSK1[6]	28 : GCLK_SERCOM7_CORE	WRCTRL.PERID[30]	CHCTRLBk.TRIG = 0x19 : RX CHCTRLBk.TRIG = 0x20 : TX	VDDREG

Table 34-6. SERCOM USART Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index: Source	MCLK AXI/APB BUS Clocks	GCLK PCHCTRLn Peripheral Channel Index: Clock Name	PAC Peripheral Peripheral Identifier (PAC.WRCTRL)	DMA Trigger Index: Source	Power Domain
SERCOM8	0x4500_2000	111 : ERROR 112 : RXBRK 113 : DRE 114 : TXC 115 : RXC 116 : RXS 117 : CTSIC	CLKMSK1[7]	29 : GCLK_SERCOM8_CORE	WRCTRL.PERID[31]	CHCTRLBk.TRIG = 0x21 : RX CHCTRLBk.TRIG = 0x22 : TX	VDDREG
SERCOM9	0x4500_4000	118 : ERROR 119 : RXBRK 120 : DRE 121 : TXC 122 : RXC 123 : RXS 124 : CTSIC	CLKMSK1[8]	30 : GCLK_SERCOM9_CORE	WRCTRL.PERID[32]	CHCTRLBk.TRIG = 0x23 : RX CHCTRLBk.TRIG = 0x24 : TX	VDDREG

34.5.7 SERCOM USART Functional Description

34.5.7.1 Principles of Operation

The USART uses the following lines for data transfer:

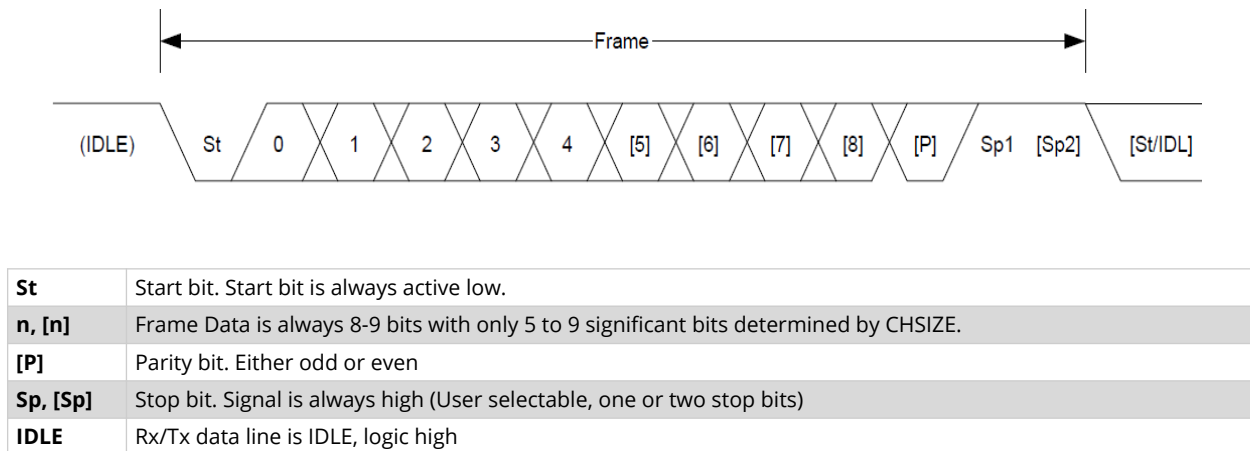
- RxD for receiving
- TxD for transmitting
- XCK/TE: XCK for the transmission clock in synchronous operation
TE for the transmit Enable in RS-485 asynchronous operation

USART data transfer is frame based. A serial frame consists of: the following

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- None, even, or odd parity bit
- 1 or 2 stop bits

A frame starts with the Start bit followed by one character of Data bits. If enabled, the parity bit is inserted after the Data bits and before the first Stop bit. After the stop bits of a frame, either the next frame can follow immediately, or the communication line can return to the Idle (high) state. The receiver clock resynchronizes to the falling edge of each start bit. The following figure illustrates the possible frame formats. Brackets denote optional bits.

Figure 34-9. Frame Formats



Note: For character Data Bits sizes defined by, USART - CTRLA.CHSIZE, data bits sizes ≤ 8 bits, 8 bits are always transmitted. The significant bits defined by CHSIZE plus padded "0's" in any remaining bits of the byte, (i.e., #padded "0" bits = $(8 - \text{CHSIZE bits})$). If CHSIZE is 9 bits, then all 9 bits with no padded zeros make up the frame.

34.5.7.2 SERCOM USART Basic Operation

34.5.7.2.1 Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (USART - CTRLA.ENABLE = 0):

- The Control A register (USART - CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- The Control B register (USART - CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- The Baud register (BAUD).

When the USART is enabled or is being enabled (USART - CTRLA.ENABLE = 1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be executed after disabling is completed.

Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before enabling the USART, it must be configured using the following steps:

1. Select either external (USART - CTRLA.MODE = 0x0) or internal clock (USART - CTRLA.MODE = 0x1).
2. Select either Asynchronous (0) or Synchronous (1) Communication mode by writing the Communication Mode.
3. Bit in the USART - CTRLA.CMODE register.
4. Select pin for receive data by writing the Receive Data Pinout value in the USART - CTRLA.RXPO register.
5. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the USART - CTRLA.TXPO register.
6. Configure the Character Size field in the USART - CTRLB.CHSIZE register for character size.
7. Set the Data Order bit in the USART - CTRLA.DORD register to determine MSB or LSB-first data transmission.

8. To use parity mode:
 - a. Enable Parity mode by writing 0x1 to the Frame Format field in the USART - CTRLA.FORM register.
 - b. Configure the Parity Mode bit in the USART - CTRLB.PMODE register for even or odd parity.
9. Configure the number of stop bits in the Stop Bit Mode bit in the USART - CTRLB.SBMODE register.
10. When using an internal clock, USART - CTRLA.MODE = 0x1, write the Baud register (BAUD) to generate the desired baud rate.
11. Enable the transmitter and receiver by writing '1' to the USART - CTRLB.TXEN and USART - CTRLB.RXEN enable bits in the USART - CTRLB register.

34.5.7.2.2 SERCOM USART Clock Generation and Selection

For both Synchronous and Asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.

The Synchronous mode is selected by writing to the Communication Mode bit in the USART - CTRLA.CMODE = 1 register bit.

Asynchronous mode is selected by writing USART - CTRLA.CMODE = 0.

The internal clock source is selected by writing USART - CTRLA.MODE = 1 register bit.

The external clock source is selected by writing USART - CTRLA.MODE = 0 register bit.

The SERCOM Baud Rate Generator is configured as shown in *Synchronous Mode XCK Timing*.



Important:

In Asynchronous mode (USART - CTRLA.CMODE = 0), the 16-bit Baud register value is used. Refer to the *Simplified USART Asynchronous Mode Block Diagram*.

In Synchronous mode (USART - CTRLA.CMODE = 1), the eight LSBs of the Baud register are used. Refer to the *Simplified USART Synchronous Mode Block Diagram*.

Refer to the SERCOM.BAUD.BAUD register/bit descriptions for details on configuring the baud rate.

34.5.7.2.3 SERCOM USART Synchronous Clock Operation

In Synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

- When CTRLA.CPOL is '0', the data will be changed on the rising edge of XCK and sampled on the falling edge of XCK.
- When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK and sampled on the rising edge of XCK.

When the external clock input is provided through XCK (CTRLA.MODE = 0x0), the RX and TX shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to $1 / [(3 * T_{RISE} I/O) + (3 * T_{FALL} I/O)]$. In Synchronous mode, no matter XCK clock is input or output, is always continuous, regardless of if no data is being transmitted or received.

Figure 34-10. Synchronous Mode XCK Timing

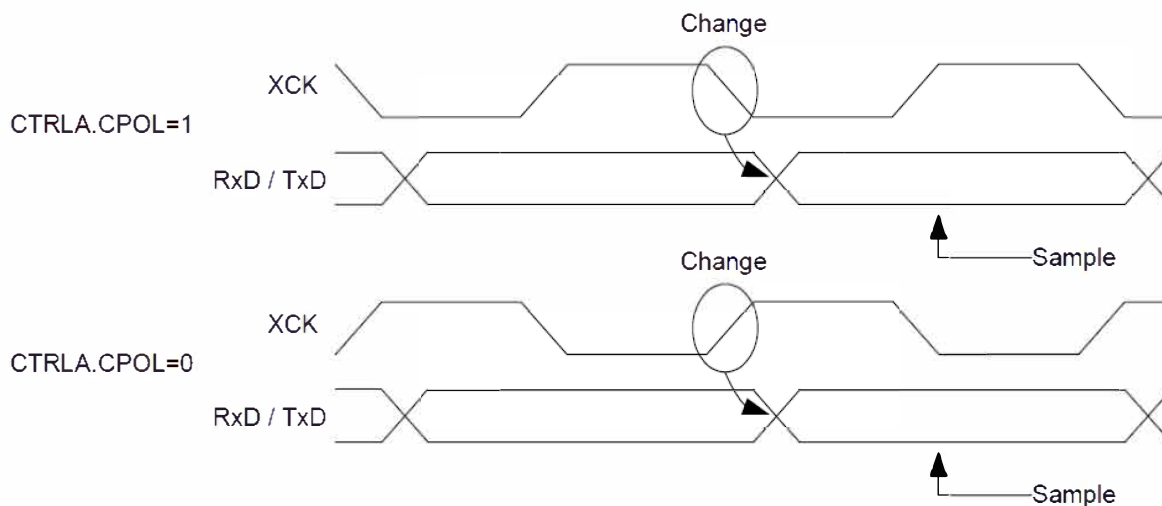
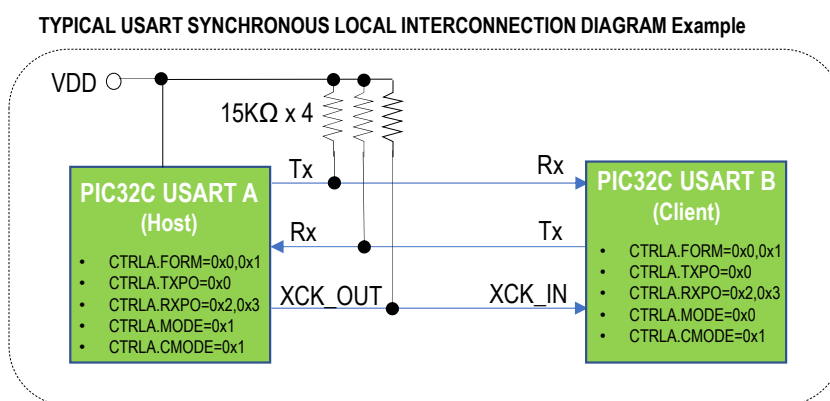


Figure 34-11. Typical USART Synchronous Use Model Example



34.5.7.2.4 SERCOM USART Tx/Rx Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

34.5.7.2.5 SERCOM USART Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the Shift register when the Shift register is empty and ready to send a new frame. After the Shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including Stop bit has been transmitted (both the TX buffer and shift register are empty) and no new data was written to DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt will be generated.

Depending on the value of CTRLC.FIFOEN, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates either:

1. If CTRLC.FIFOEN = 1, then at a minimum, (CTRLC.TXTRHOLD), FIFO threshold locations are empty and ready for new data, INTFLAG.DRE = 1.

2. If CTRLC.FIFOEN = 0, then the DATA register is ready to accept new TxDATA, INTFLAG.DRE = 1.

 **Important:** In either case DATA register should only be written to when INTFLAG.DRE = 1.

Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit register (CTRLB.TXEN).

Disabling the transmitter will not take effect until after any ongoing and pending transmissions are completed, INTFLAG.TXC = 1.

34.5.7.2.6 SERCOM USART Data Reception

The receiver accepts data when a valid Start bit is detected. Each bit following the Start bit will be sampled in asynchronous mode according to ((CTRLA.SAMPR = 3, 8, 16) * baud rate) or in synchronous mode according to XCK clock and shifted into the receive Shift register until the first Stop bit of a frame is received. The second Stop bit will be ignored by the receiver. When the first Stop bit is received and a complete serial frame is present in the Receive Shift register, the contents of the Shift register will be moved into the receive buffer.

If CTRLC.FIFOEN = 1, when the number of bytes present in the FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD setting the Receive Complete Interrupt flag in the Interrupt Flag Status register, (INTFLAG.RXC), will be set: An optional interrupt will be generated if INTENSET.RXC = 1. The user can continue to read the DATA register as long as INTFLAG.RXC = 1 until the FIFO is empty and INTFLAG.RXC = 0.

If CTRLC.FIFOEN = 0, the Receive Complete Interrupt flag in the Interrupt Flag Status register, (INTFLAG.RXC), will be set: An optional interrupt will be generated if INTENSET.RXC = 1. The user can read the received data from the DATA register until INTFLAG.RXC = 0.

Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level receive buffer, and data from ongoing receptions will be lost.

Error Bits

The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be set until it is cleared by writing '1' to it. These bits are also cleared automatically when the receiver is disabled.

On any error, (STATUS.ERROR = 1), the USART transmit and receive operations are halted until the DATA register is read and emptied, INTFLAG.RXC = 0, by the user software followed after by the user clearing all the corresponding STATUS register error bits. Subsequent data received after the initial error is lost until the error conditions are resolved and normal operation can resume.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

- When CTRLA.IBON = 1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading the DATA register, until the Receiver Complete Interrupt flag (INTFLAG.RXC) is cleared.
- When CTRLA.IBON = 0, the Buffer Overflow condition is attending data through the receive FIFO. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

Asynchronous Data Reception

The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock. The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

Asynchronous Operational Range

The operational range of the asynchronous reception depends on the difference between the internal baud-rate clock frequency of the receiving device relative to the internal baud rate frequency of the remote transmitting device, (i.e., baud rate error). The maximum baud rate error should exceed $\pm 1.5\%$ for any individual serial device in the network for a total error between devices not to exceed $\pm 3\%$. The $\pm 3\%$ maximum error assumes that the receiver and transmitter equally divide the $\pm 3\%$ maximum total error for $\pm 1.5\%$ for each USART device.

$$\text{Error \%} = [(1 - (\text{Expected Baud Rate} / \text{Actual Baud Rate})) * 100]$$

34.5.7.3 SERCOM USART Additional Features

34.5.7.3.1 Parity

Even or odd parity can be selected for error checking by setting CTRLA.FORM=0x1 in the Frame Format bit field of the Control A register.

If *even parity* is selected (CTRLB.PMODE = 0), the Parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

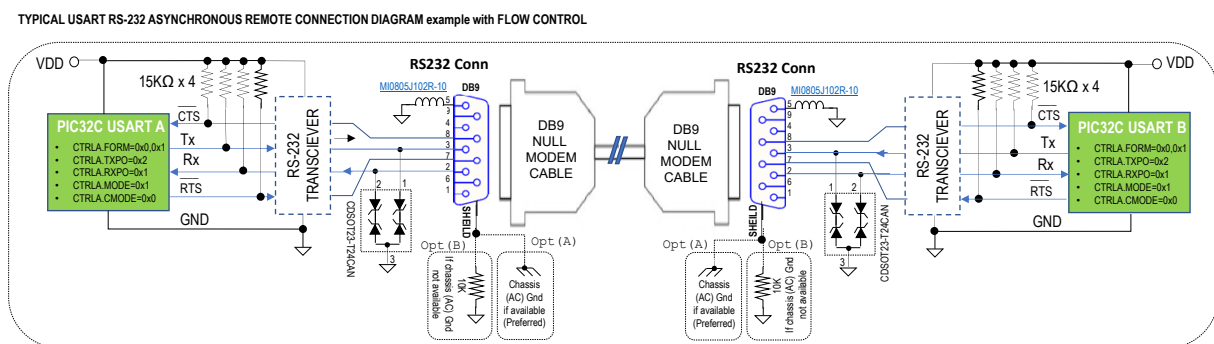
If *odd parity* is selected (CTRLB.PMODE = 1), the Parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the Parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

34.5.7.4 SERCOM USART Hardware Handshaking

The USART features a hardware handshaking flow control option, implemented by cross connecting the RTS pin of a receiving device to the CTS pin of transmitting device as shown in the following figure.

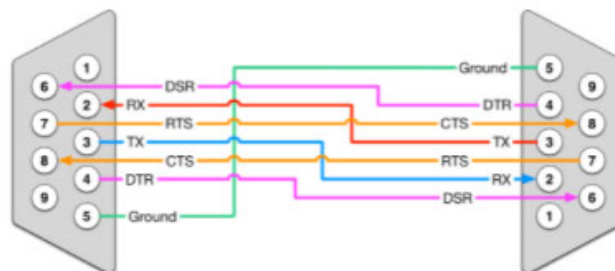
Figure 34-12. Typical RS-232 Use Example with a Remote Device Hardware Handshaking (Flow Control)



Notes:

1. The recommended ground inductor should be a power line ferrite $\geq 680 \Omega$ @ 100 MHz w/DCR $\leq 0.15 \Omega$.
2. The TVS, Transient Voltage Suppressor, should be a bi-directional Zener w/Reverse Standoff voltage rating $> 10\%$ of the maximum signal operating levels, and a Peak Pulse current rating ≥ 10 amps.

Figure 34-13. Null Modem Cable Wiring Example



- **RTS:** Request to Send Output
- **Logic low:** Receiving device is ready to receive data
- **Logic high:** Receiving device not ready to receive data
- **CTS:** Clear to Send Input
- **Logic low:** Indication to transmitting device that it's clear to transmit
- **Logic high:** Indication to transmitting device to stop transmitting
- Hardware handshaking is only available in the following configuration:
 - USART with internal clock (CTRLA.MODE=1)
 - Asynchronous mode (CTRLA.CMODE=0)
 - Flow control pinout (CTRLA.TXPO=2)

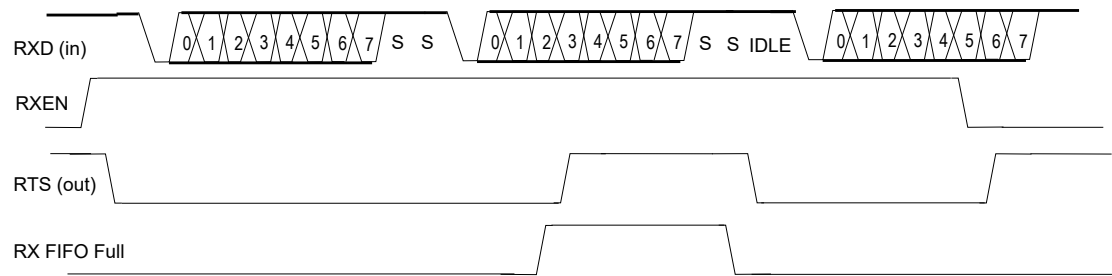
When using Flow Control, the receiving device will drive the RTS pin inactive high whenever any of the following conditions are met:

- CTRLB.RXEN=0
- If FIFOEN=1, and the receive FIFO is full
- FIFOEN=0, and the receive RxDATA, (DATA register), is full and INTFLAG.RXC=1

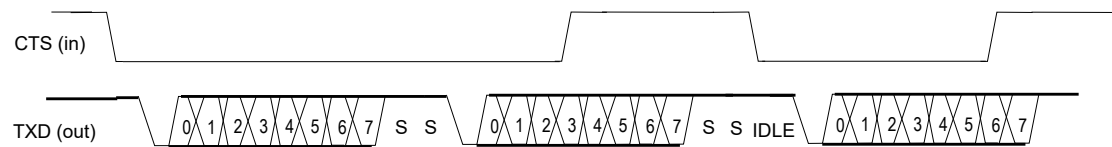
When any of the conditions above are met, RTS output from the receiving device goes inactive, (i.e., logic high). This notifies the remote transmitting device through the CTS input pin to stop the transfer after any ongoing transmission that may already be in progress to stop. Any transfer in progress when RTS goes inactive, will complete with the incoming data stored in the RX shift register until if ,FIFOEN = 1, the receive FIFO is no longer full or if FIFOEN = 0 the DATA register is empty, (i.e., INTFLAG.RXC = 0). Enabling and disabling the receiver by writing to CTRLB.RXEN will clear/set respectively the RTS pin after a synchronization delay.

Figure 34-14. Receiver/Transmitter Behavior when Operating with Hardware Handshaking (Flow Control)

Receiving Device



Transmitting Device



The current CTS pin Status is reflected in the STATUS register (STATUS.CTS). Character transmission will resume only if STATUS.CTS = 0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting. Disabling the receiver, (CTRLB.RXEN = 0,) will disable the receiver, flush the two-level receive buffer, and data from any ongoing receptions will be lost.

34.5.7.5 DMA, Interrupts, and Events

Table 34-7. DMA Module Request for SERCOM USART

CONDITION	REQUEST		
	DMA	INTERRUPT	EVENT
Standard (DRE): Data Register Empty FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty	Yes (Request cleared when data is written)	Yes	N/A
Standard (RXC): Receive Complete FIFO (RXC): at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed.	Yes (Request cleared when data is read)	Yes	
Standard (TXC): Transmit Complete FIFO (TXC): Transmit Complete and TX FIFO is empty	N/A	Yes	
Receive Start (RXS)	N/A	Yes	
Clear to Send Input Change (CTSIC)	N/A	Yes	
Receive Break (RXBRK)	N/A	Yes	
Error (ERROR)	N/A	Yes	

34.5.7.5.1 DMA Operation

The USART generates the following DMA requests:

- **Data received (RXC):** The request is set when data is available in the receive FIFO or if at least RXTRHOLD data are available in the RX FIFO when FIFO operation is enabled. The request is cleared when DATA is read.
- **Data transmit register empty (DRE):** The request is set when the transmit buffer (TX DATA) is empty or if at least TXTRHOLD data locations are empty in the TX FIFO, when FIFO operation is enabled. The request is cleared when DATA is written.

34.5.7.5.2 Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake-up the device from any Sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET) and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the Interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled, or the USART is reset. For details on clearing Interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to the *Nested Vector Interrupt Controller* for details.

34.5.7.6 Sleep Mode Operation

The behavior in Sleep mode is depending on the clock source and the Run Standby bit in the CTRLA.RUNSTDBY register:

Table 34-8. Sleep Mode Behavior

Clock Source	CTRLA. RUNSTDBY	Wake Up Source	Comment
Internal, GCLK_SERCOMx_CORE	1	Any Interrupt	Any interrupt can wake-up the device
External, XCK SERCOM PAD[1,3]		INTFLAG RXC	The Receive Complete interrupt(s) can wake-up the device
Internal, GCLK_SERCOMx_CORE	0	INTFLAG RXC	Internal clock will be disabled, after any ongoing transfer is completed. The Receive Complete interrupt(s) can wake-up the device.
External, XCK SERCOM PAD[1,3]		None	External clock will be disconnected after any ongoing transfer is completed then all reception will be stopped.



Important: As described in the *Start of Frame Detection* section, it is recommended the user select a fast start-up internal RC oscillator for the GCLK_SERCOMn_CORE clock for fast wake from sleep mode if desired, otherwise the frame data from the initial Start-of-Frame detection event may get corrupted and any succeeding frames until the oscillator clock has started up and stable due to baud rate mismatch issues.

34.5.7.7 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers must be synchronized when written or read.

The following bits are synchronized when written:

- The Software Reset bit in the Control A register (CTRLA.SWRST)
- The Enable bit in the Control A register (CTRLA.ENABLE)
- The Receiver Enable bit in the Control B register (CTRLB.RXEN)
- The Transmitter Enable bit in the Control B register (CTRLB.TXEN)

Notes:

1. Required write synchronization is denoted by the "Write-Synchronized" property in the register description. If a write-synchronized register is written while a synchronization is ongoing, an APB error will be generated.
2. CTRLB.RXEN is write-synchronized somewhat differently. For additional information, refer to the CTRLB.RXEN bit description.

34.6 SERCOM Serial Peripheral Interface (SPI)

34.6.1 Overview

The Serial Peripheral Interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as shown in the Block Diagram. SPI communications consists of a Host and at least one Client. Each uses a separate SPI module containing a Shift register, a transmit buffer, and a receive buffer.

The SPI Host uses the SERCOM Baud Rate Generator. The SPI Client can use the SERCOM address match logic.

Note that Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM and presented to help in explaining how the module functions.

34.6.2 Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, SS)
- One-level transmit buffer, two-level receive buffer
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on the MISO or MOSI pin
- Selectable LSB or MSB-first data transfer
- Can be used with DMA
- 0-bit Extension for better system bus utilization
- Framed SPI protocol support in both Host and Client operating mode, with hardware controlled FSYNC
- Internal FIFO (See Table below)
- Host operation:
 - Serial clock speed, $f_{SCK}=1/t_{SCK}^1$
 - 8-bit clock generator
 - Hardware controlled SS
 - Optional inter-character spacing
- Client Operation:
 - Serial clock speed, $f_{SCK}=1/t_{SSCK}^1$
 - Optional 8-bit address match operation
 - Operation in all sleep modes

- Wake on SS transition

Note:

1. For t_{SCK} and t_{SSCK} values, refer to SPI Timing Characteristics.

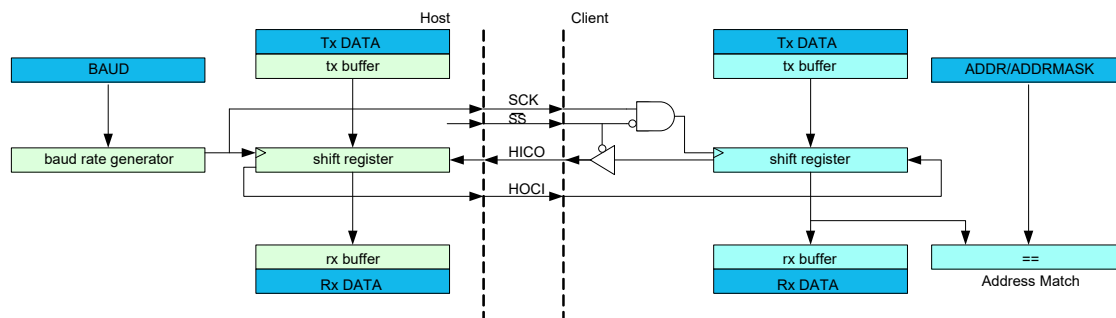
Table 34-9. FIFO Buffer Configuration

FIFO Status	Transmit Buffer(s)	Receive Buffer(s)
Disabled or Not Implemented	Single write buffer.	A two-level receive buffer.
Enabled	0 write FIFO buffer.	0 receive FIFO buffer.

In addition, the SPI Host uses the SERCOM baud-rate generator, while the SPI Client can use the SERCOM address match logic.

34.6.3 Block Diagram

Figure 34-15. Full-Duplex SPI Host Client Interconnection



When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. If the receiver is disabled, the data input pin can be used for other purposes. In Host mode, the client select line ($\overline{SS}$) is hardware controlled when the Host Client Select Enable bit in the Control B register (CTRLB.MSEN) is '1'.

Table 34-10. SPI Pin Configuration

Pin	Host SPI	Client SPI
MOSI	Output	Input
MISO	Input	Output
SCK	Output	Input
$\overline{SS}$ / FSYNC	Output (CTRLB.MSEN=1) and (CTRLC.FRMEN = 0)	Input (CTRLC.FRMEN = 0)
	Output (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 0)	Output (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 0)
	Input (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 1)	Input (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 1)

The combined configuration of PORT, the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals in the table above.

34.6.4 Functional Description

34.6.4.1 Principle of Operation

The SPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral devices.

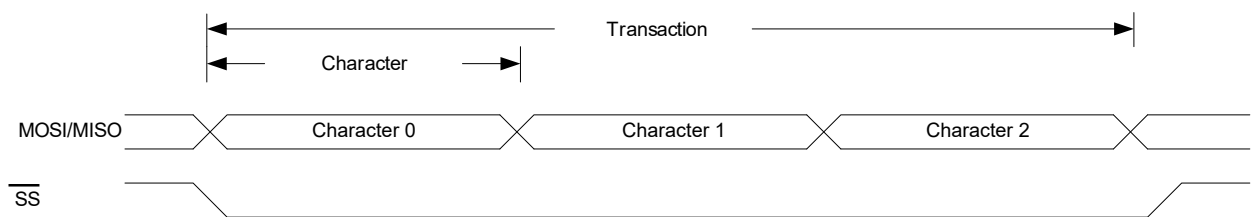
The SPI can operate as host or client. As host, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 16 bits.

Figure 34-16. SPI Transaction Format



The SPI host must pull the client select line ($\overline{SS}$) of the desired client low to initiate a transaction. The host and client prepare data to send via their respective Shift registers, and the host generates the serial clock on the SCK line.

Data are always shifted from host to client on the Host Output Client Input line (MOSI); data is shifted from client to host on the Host Input Client Output line (MISO).

Each time a character is shifted out from the host, a character will be shifted out from the client simultaneously. To signal the end of a transaction, the host will pull the $\overline{SS}$ line high.

34.6.4.2 Basic Operation

34.6.4.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE=1), any writing to these registers will be discarded

When the SPI is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in host/client operation in the Operating Mode bit group in the CTRLD register (CTRLD.MODE= 0x2 or 0x3).
2. Select Transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLD register (CTRLD.CPOL and CTRLD.CPHA) if desired.

3. Select the Frame Format value in the CTRLD register (CTRLD.FORM).
4. Configure the Data In Pinout field in the Control D register (CTRLD.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control D register (CTRLD.DOPO) for SERCOM pads of the transmitter.
6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLD register (CTRLD.DORD) for data direction.
8. If the SPI is used in Host mode:
 - Select the desired baud rate by writing to the Baud register (BAUD).
 - If Hardware $\overline{SS}$ control is required, write '1' to the Host Client Select Enable bit in CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

34.6.4.2.2 Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

34.6.4.2.3 Clock and Clock Generation

Clock

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Device Main Clock Controller.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Device Generic Clock Controller before using the SPI.

Note: GCLK_SERCOMn_CORE" is the only clock the user should select for the SERCOM in SPI mode.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization to the clock domains. Refer to Synchronization for further details.

Clock Generation

In SPI host operation (CTRLD.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM baudrate generator.

In SPI mode, the baud-rate generator is set to Synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the Shift register. Refer to Clock Generation – Baud-Rate Generator for more details.

In SPI client operation (CTRLD.MODE is 0x2), the clock is provided by an external master on the SCK pin. This clock is used to directly clock the SPI Shift register.

34.6.4.2.4 Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing to the DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

34.6.4.2.5 Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI Data Transfer modes are shown in the SPI Transfer Modes table, and the SPI Transfer Modes figure.

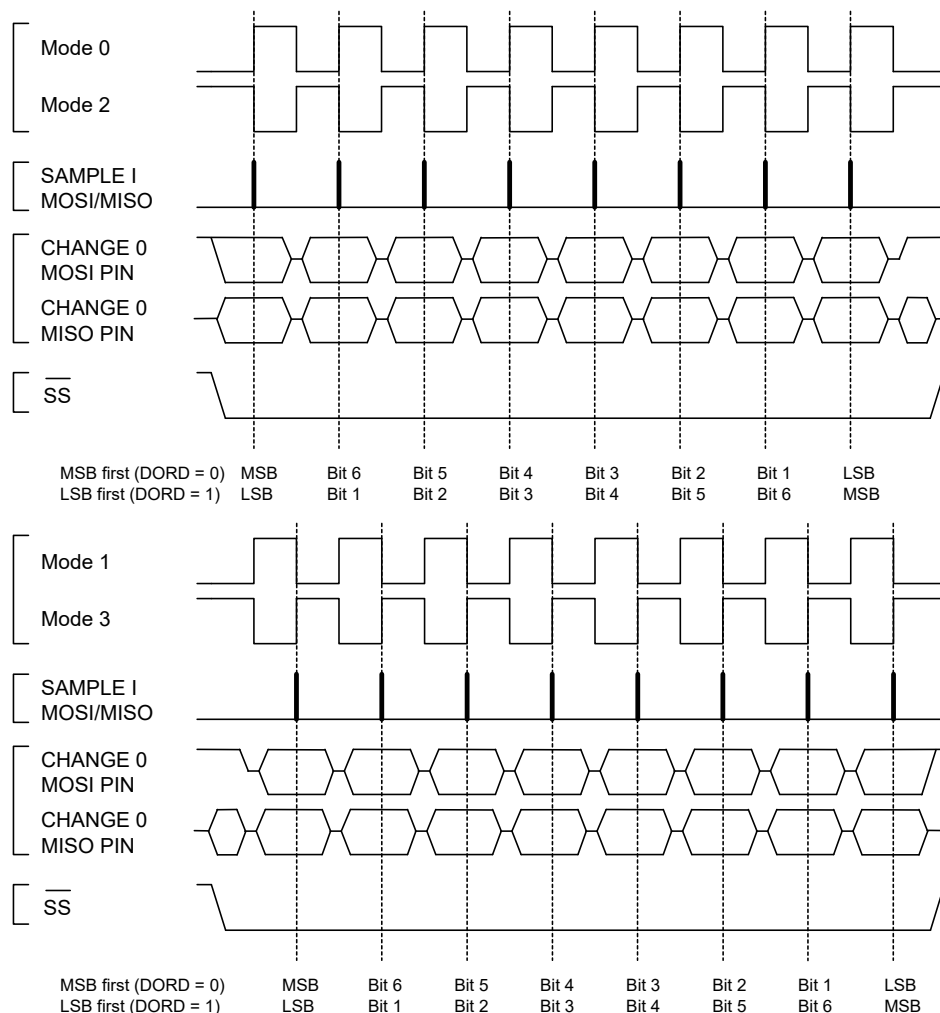
SCK phase is configured by the Clock Phase bit in the CTRLD register (CTRLD.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLD register (CTRLD.CPOL). Data bits are shifted out and latched in, on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

Table 34-11. SPI Transfer Modes

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

Note: Leading edge is the first clock edge in a clock cycle.
Trailing edge is the second clock edge in a clock cycle.

Figure 34-17. SPI Transfer Modes



34.6.4.2.6 Transferring Data

Host Transferring Data

In Host mode (CTRLD.MODE=0x3), when Host Client Enable Select (CTRLB.MSEN) is '1', hardware will control the $\overline{SS}$ line.

When Host Client Select Enable (CTRLB.MSSEN) is '0', the SS line must be configured as an output. SS can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the SS line low.

For configurations where the FIFO is disabled or not implemented, when writing a character to the Data register (DATA), the character will be transferred to the Shift register. Once the content of TxDATA has been transferred to the Shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA. If the FIFO is enabled, INTFLAG.DRE will be set when at least FIFO threshold (CTRLC.TXTRHOLD) locations in TX FIFO are empty. Characters can be written as long as the FIFO is not full.

Each time one character is shifted out from the Host, another character will be shifted in from the Client simultaneously. If the receiver is enabled (CTRLC.RXEN=1), the contents of the Shift register will be transferred to the receive buffer. The transfer takes place in the same clock cycle as the last data bit is shifted in.

For configurations where the FIFO is disabled or not implemented, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. For configuration where the FIFO is enabled, RXC is set when at least RXTRHOLD data are available in the RX FIFO. In both cases, the received data can be retrieved by reading DATA.

For configurations where the FIFO is disabled or not implemented, when the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. For configuration where the FIFO is enabled, TXC is set when the transmit is complete and the TX FIFO is empty. When the transaction is finished, the Host must pull the SS line high to notify the Client. If Host Client Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the SS line high.

Client Transferring Data

In Client mode (CTRLD.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the SS pin is pulled high. Software may update the contents of Tx buffer at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When SS is pulled low and SCK is running, the Client will sample and shift out data according to the Transaction mode set.

For configurations where the FIFO is disabled or not implemented, when the content of DATA has been loaded into the Shift register, INTFLAG.DRE will be set, and new data can be written to DATA. If the FIFO is enabled, INTFLAG.DRE will be set when at least FIFO threshold (CTRLC.TXTRHOLD) locations in TX FIFO are empty. When DATA or FIFO threshold locations are empty respectively, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

Similar to the Host, the Client will receive one character for each character transmitted. A character will be transferred into the receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set. For configurations where the FIFO is disabled or not implemented, the RXC is set when a character reception is complete. If the FIFO is enabled, the RXC is set when at least RX threshold (CTRLC.RXTRHOLD) data are received.

When the Host pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set.

After DATA is written it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the Shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. Refer to Preloading of the Client Shift Register.

When transmitting several characters in one SPI transaction, the data has to be written into the DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character will be transmitted.

34.6.4.2.7 Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate Buffer Overflow Notification bit in the Control D register (CTRLD.IBON):

1. If CTRLD.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA until the receiver complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.
2. If CTRLD.IBON=0, the Buffer Overflow condition travels with data through the receive FIFO. After the received data is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

34.6.4.3 Additional Features

34.6.4.3.1 Address Recognition

When the SPI is configured for client operation (CTRLA.MODE=0x2) with address recognition (CTRLD.FORM is 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled, and the transaction is processed. If the device is in Sleep mode, an address match can wake-up the device in order to process the transaction.

If there is no match, the complete transaction is ignored.

If a 0-bit frame format is selected, only the lower 8 bits of the Shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN=0) in order to use this mode.

34.6.4.3.2 Preloading of the Client Shift Register

When starting a transaction, the client will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

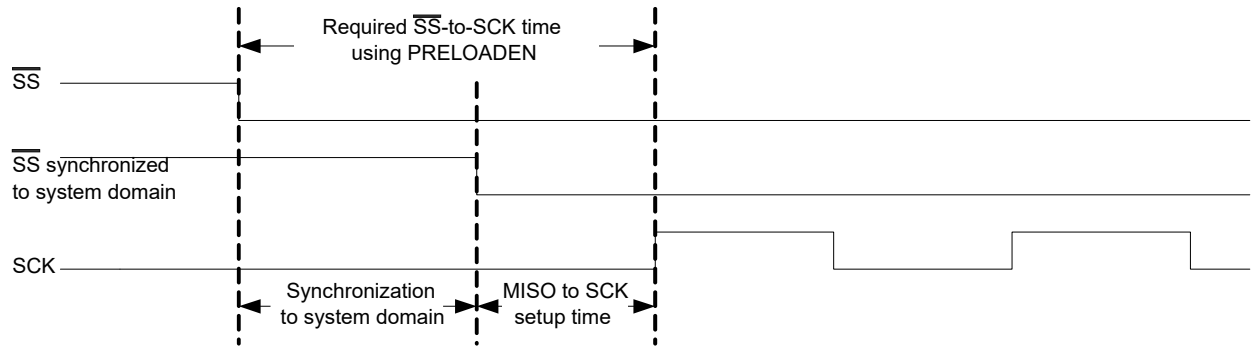
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading. See also the *Electrical Characteristics* chapters for timing details.

Preloading is enabled by writing '1' to the Client Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

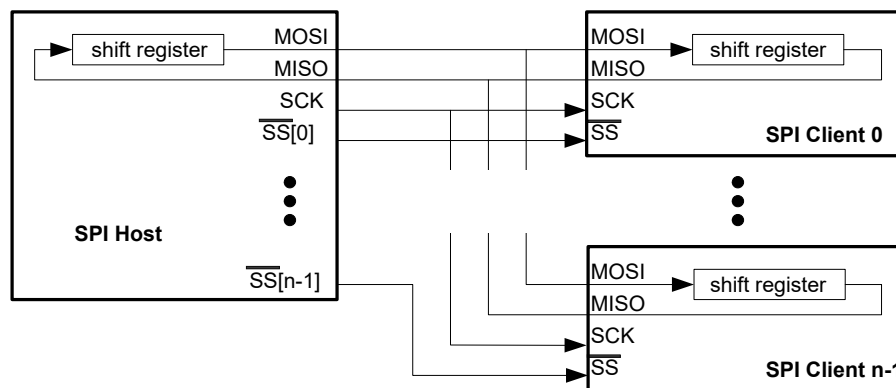
Figure 34-18. Timing Using Preloading



34.6.4.3.3 Host with Several Clients

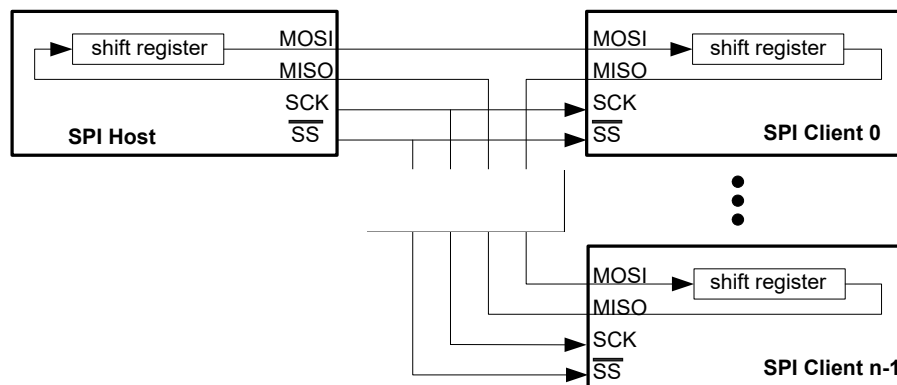
Host with multiple clients in parallel is only available when Host Client Select Enable (CTRLB.MSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI clients, a SPI host can use general purpose I/O pins to control the $\overline{SS}$ line to each of the clients on the bus, as shown in Multiple Slaves in Parallel. In this configuration, the single selected SPI client will drive an internal tri-stated MISO line.

Figure 34-19. Multiple Clients in Parallel



Another configuration is multiple clients in series, as in Multiple Clients in Series. In this configuration, all n attached clients are connected in series. A common $\overline{SS}$ line is provided to all clients, enabling them simultaneously. The host must shift n characters for a complete transaction. Depending on the Host Client Select Enable bit (CTRLB.MSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

Figure 34-20. Multiple Clients in Series



34.6.4.3.4 Loopback Mode

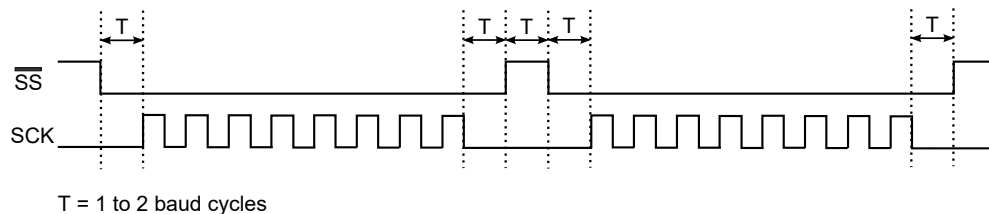
For Loop-back mode, configure the Data In Pinout (CTRLD.DIPO) and Data Out Pinout (CTRLD.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

34.6.4.3.5 Hardware Controlled $\overline{SS}$

In Host mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Host Client Select Enable (CTRLB.MSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI Transfer mode.

Figure 34-21. Hardware Controlled $\overline{SS}$



When CTRLB.MSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

34.6.4.3.6 Client Select Low Detection

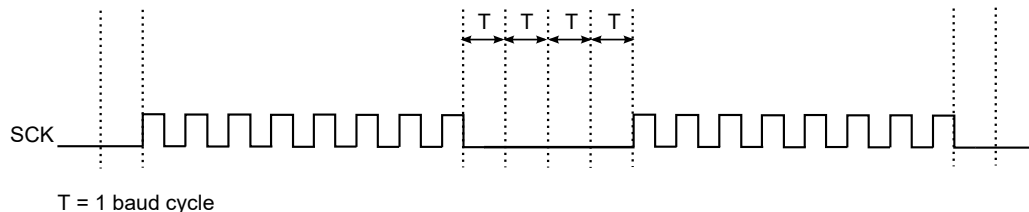
In Client mode, the SPI can wake the CPU when the client select ($\overline{SS}$) goes low. When the Client Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Client Select Low Interrupt flag (INTFLAG.SSL) and the device will wake-up if applicable.

34.6.4.3.7 Host Inter-Character Spacing

When configured as host, inter-character spacing can be increased by writing a non-zero value to the Inter-Character Spacing bit field in the Control C register (CTRLC.ICSPACE). When non-zero, CTRLC.ICSPACE represents the minimum number of baud cycles that the SCK clock line does not toggle and the next character is stalled.

The following figure gives an example for CTRLC.ICSPACE=4; In this case, the SCK is inactive for 4 baud cycles.

Figure 34-22. Four Cycle Inter-Character Spacing Example



34.6.4.3.8 0-bit Extension

For better system bus utilization, 0-bit data receive and transmit can be enabled by writing to the Data 0-bit bit field in the Control C register (CTRLC.DATA32B=1). When enabled, write and read transaction to/from the DATA register are 0 bit in size.

If frames are not multiples of 0 Bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The figure below shows the order of transmit and receive when using 0-bit mode. Bytes are transmitted, or received and stored, in order from 0 to 3.

Only 0-bit character size is supported.

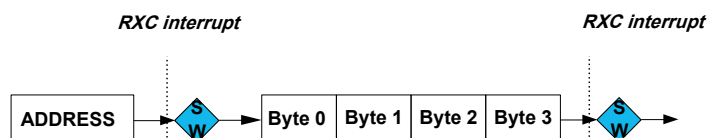
Figure 34-23. 0-bit Extension Byte Ordering



0-bit Extension Client Operation

The following figure shows a transaction with 0-bit Extension enabled (CTRLC.DATA32B=1). When address recognition is enabled (CTRLD.FORM=0x2) and there is an address match, the address is loaded into the FIFO as Byte zero and data begins with Byte 1. INTFLAGS.RXC will then be raised for every 0 Bytes transferred. For transmit, there is a 0-bit holding buffer in the core domain. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised, so that the next 0 bits can be written to the DATA register.

Figure 34-24. 0-bit Extension Client Operation



When utilizing the length counter, the LENGTH register must be written before the frame begins. If the frame length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN Bytes, the Length Error Status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 0 Bytes, the final INTFLAG.RXC interrupt will be raised when the last Byte is received.

The length count is based on the received Bytes, or the number of clocks if the receiver is not enabled. If pre-loading is disabled and DATA is written to for transmit before SCK starts, transmitted data will be delayed by one Byte, but the length counter will still increment for the first (empty) Byte transmission. When the counter reaches LENGTH.LEN, the internal length counter, Rx Byte counter, and Tx Byte counter are reset. If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

If there is a Length Error (STATUS.LENERR), the remaining Bytes in the length will be transmitted at the beginning of the next frame. If this is not desired, the SERCOM must be disabled and re-enabled in order to flush the Tx and Rx pipelines.

Writing the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not configured and a frame is not a multiple of 4 Bytes (while $\overline{SS}$ is low), the remainder will be transmitted in the next frame.

0-bit Extension Host Operation

When using the SPI configured as Host, the Length and the Length Enable bit fields (LENGTH.LEN and LENGTH.LENEN) must be written before the frame begins. When LENGTH.LENEN is written to '1', the value of LENGTH.LEN determines the number of data bytes in the transaction from 1 to 255.

For receive data, INTFLAG.RXC is raised every 0 Bytes received. If LENGTH.LEN is not a multiple of 0 Bytes, the final INTFLAG.RXC is set when the final byte is received.

For transmit, there is a holding buffer for the 0-bit data in the core domain. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised so that the next 0 bits can be written to the DATA register.

If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

34.6.4.3.9 Framed SPI Operation

The SPI supports Framed SPI protocol while operating in either Host or Client mode. The CTRLC.FRMEN bit enables Framed SPI modes and causes the SS pin to be used as a frame synchronization (FSYNC) input or output pin. The state of CTRLB.MSEN bit is then ignored.

Unlike in normal SPI mode, the serial clock is continuous (free-running) in Framed SPI mode rather than being generated only when there is data to be transmitted. The data transmission/reception starts only when the frame synchronization is generated at the FSYNC pin. The device can be either a frame host if it generates the frame sync or a frame client if it receives the frame sync at the FSYNC pin. In other words, only a frame host can generate the frame synchronization pulse.

Frame Host or Client mode is selected by clearing or setting the CTRLC.FMODE bit, respectively. The frame synchronization can be an active-high or an active-low pulse of one SCK clock duration, or frame duration, based on the CTRLC.FSPOL and CTRLC.FSLEN settings.

Irrespective of which device is a host and which is a client, a framed SPI data transfer begins as soon as the frame host generates the frame sync and writes the data to DATA. For full-duplex operation, the frame client should write to its buffer before the frame host does, in order to ensure that the data is ready at both ends when the data transfer begins.

Frame Host

The frame host mode is enabled by setting CTRLC.FMODE = HOST and CTRLC.FRMEN = 1.

If the module is enabled in host operating mode (CTRLD.MODE = 0x03), the serial clock is output at the SCK pin, regardless of whether the module is transmitting, and the FSYNC pin is driven high on the next transmit edge of the SCK clock when the Data buffer is written. Data will start transmitting on the subsequent transmit edge of the SPI clock. As long as data is available in the transmit buffer, a new frame sync is initiated after completing a transmit/receive sequence.

If the module is enabled in client operating mode (CTRLD.MODE = 0x2), the input clock at the SCK pin is continuous while the FSYNC pin and transmission are generated in the same way as in host operating mode.

If the 0-bit extension is disabled (CTRLC.DATA32B = 0), the transmit will continue as long as data are present in the internal buffer. The Transmit Complete interrupt flag is set when there are no more data to be transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new data can be stored in the internal buffer.

If the 0-bit extension is enabled (CTRLC.DATA32B = 1), and the DATA Length Counter is disabled (LENTH.LENEN = 0), the behavior is similar to the previous mode, with the exception the DATA must be 0-bit format (word).

When the DATA Length Counter is enabled (LENTH.LENEN = 1), the length of the frame is controlled by the DATA Length field in LENGTH register (LENGTH.LEN). The Transmit Complete interrupt flag is set when there are no more data to be transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new 0-bit data can be stored in the internal buffer.

A DMA trigger is generated each time there is 0-bit internal place to store a new data. As a consequence, the DMA descriptor transfer counter must be 0-bit aligned.

Figure 34-25. FMODE = HOST, FSPOL = HIGH, FSLEN = STROBE, FSES = 0

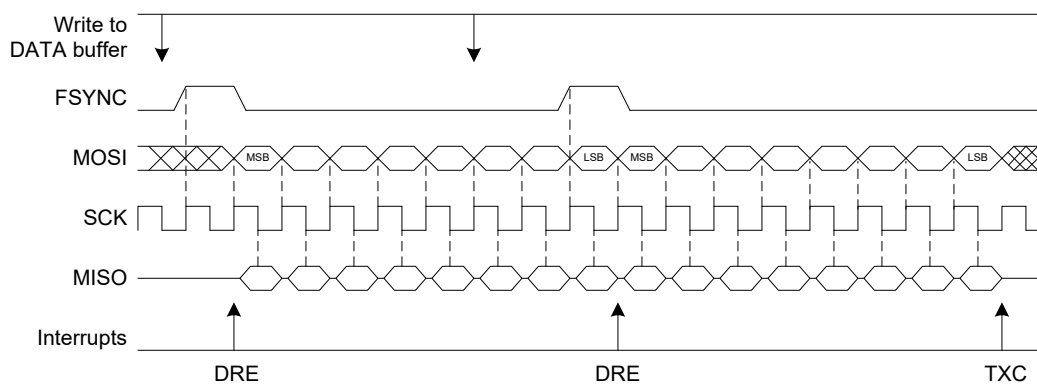


Figure 34-26. FMODE = HOST, FSPOL = HIGH, FSLEN = LEVEL, FSES = 0

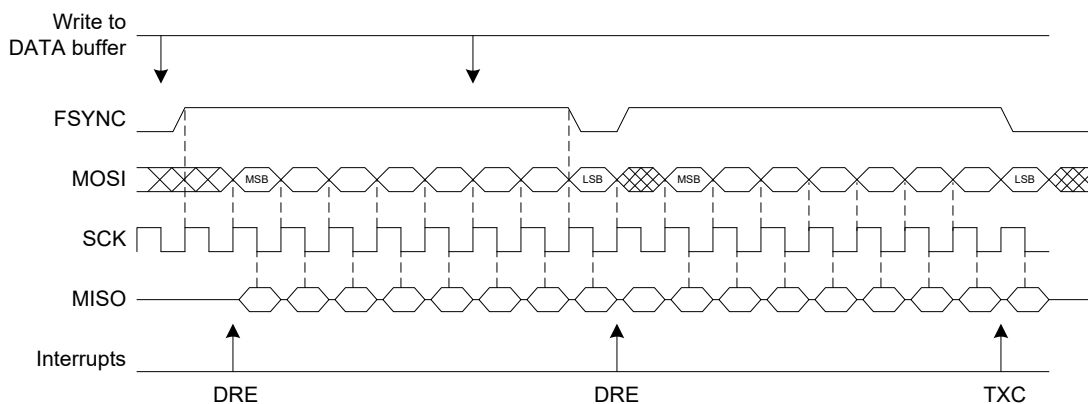


Figure 34-27. FMODE = HOST, FSPOL = HIGH, FSLEN = STROBE, FSES = 1

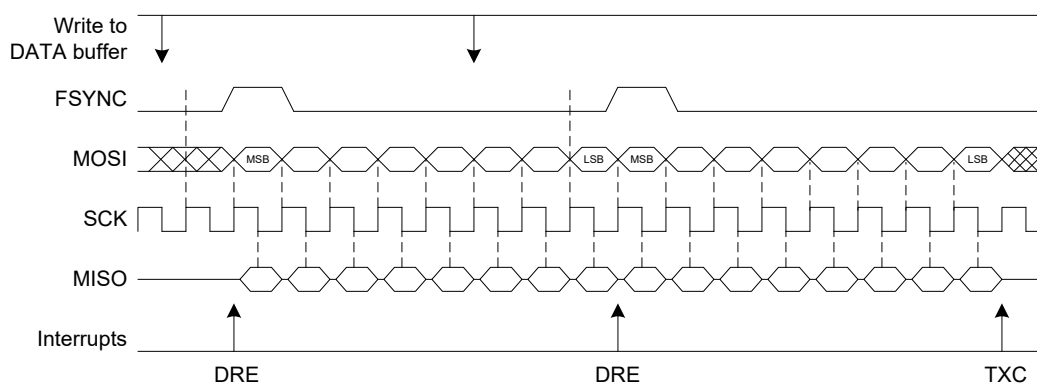
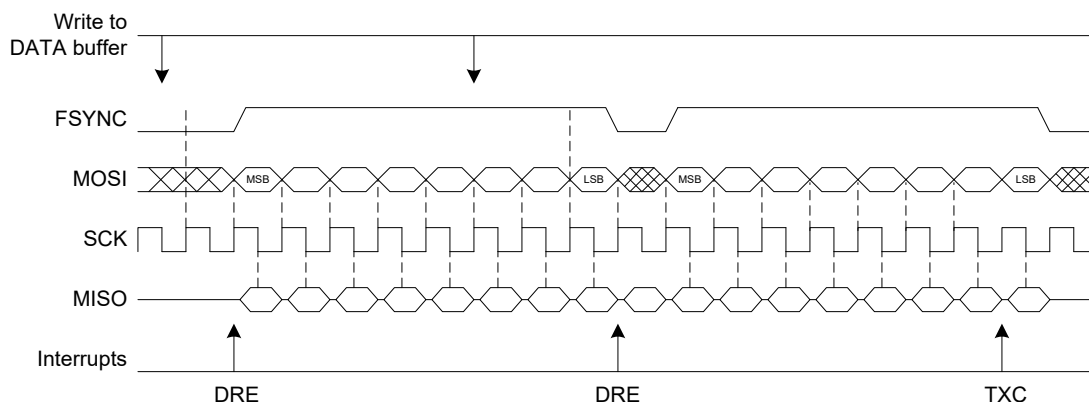


Figure 34-28. FMODE = HOST, FSPOL = HIGH, FSLEN = LEVEL, FSSES = 1



Frame Client

The frame client mode is enabled by setting CTRL.C.FMODE = CLIENT and CTRL.C.FRME = 1. In both host (CTRL.D.MODE = 0x03) and client (CTRL.D.MODE = 0x02) operating mode, the FSYNC pin is an input. The serial clock is output at the SCK pin in host operating mode, and an input in client operating mode.

When the FSYNC pin is sampled high or low (CTRL.C.FSPOL = 1/0), the data is transmitted on the subsequent transmit edge of the SPI clock. The user must make sure that the correct to-be-transmitted data is loaded into DATA buffer before the frame sync pulse is received at the FSYNC pin. If the FSYNC is detected and there is no data available in the buffer, the module will transmit zeroes until the end of the frame.

If the 0-bit extension is disabled (CTRL.C.DATA32B = 0), one byte DATA will be sent, and a new FSYNC must be received before enabling a new transmission. The Transmit Complete interrupt flag is set when the internal shift register is empty. The Data Register Empty flag is set each time a new data can be stored in the internal buffer.

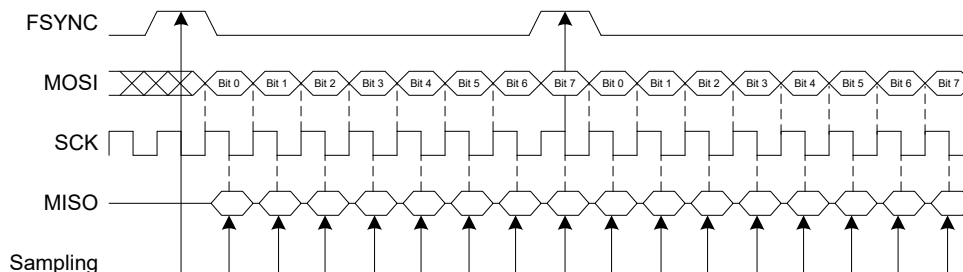
If the 0-bit extension is enabled (CTRL.C.DATA32B = 1), and the DATA Length Counter is disabled (LENTH.LENEN = 0), the behavior is similar to the previous mode, with the exception the DATA will be 0-bit format (word).

When the DATA Length Counter is enabled (LENTH.LENEN = 1), the length of the frame is controlled by the DATA Length field in LENGTH register (LENTH.LEN). The Transmit Complete interrupt flag is set when LENGTH DATA are transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new 0-bit data can be stored in the internal buffer. A new FSYNC must be received before a new transaction starts.

If the address matching is enabled, the first data after FSYNC detection will be compared with the ADDR register. If a match occurs, the client continues to receive data. If the match fails, the data is discarded and the client waits for a new FSYNC.

A DMA trigger is generated each time there is 0-bit internal place to store a new data. As a consequence, the DMA descriptor transfer counter must be 0-bit aligned.

Figure 34-29. FMODE = CLIENT, FSPOL = HIGH, FSLEN = STROBE, FSES = 0



Framed SPI Errors

If the transmit buffer is empty when the SPI must load the shift register to start (in the case of Framed Client) or continue (for either Framed Host or Client) a transfer, the SPI immediately sets the transmit underrun status bit in the STATUS (STATUS.TUR) register. If enabled, an Error interrupt flag (INTFLAG.ERROR) will be also generated.

While the SPI is in an underrun condition, the SPI transmits all zeroes until the end of the transaction (byte, word or length counter). If the SPI is a frame client, another frame sync pulse can occur before the condition clears. In this case the SPI continues to transmit zeros. If the SPI is a frame host, it then waits for software to clear the under-run condition before initiating another FSYNC regardless of the state of the DATA buffer.

When the TUR status is cleared the data present in the DATA buffer is flushed. The SPI ignores writes to the DATA after clearing the STATUS.TUR bit, until a read of STATUS register, when STATUS.TUR=0. This behavior ensures that a data service routine that is interrupted long enough to cause STATUS.TUR, cannot inadvertently start a new framed transaction after the SPI error handler has cleared the error.

For cases when software does not care or need to know about the underrun condition, CTRLC.IGNTUR = 1 provides the serial engine the ability to ignore the underrun. When an underrun occurs, the SPI still sets the corresponding STATUS and INTFLAG bits. When the SPI is either a frame client or a frame host, an underrun event still causes the SPI to transmit zeros until the end of the frame. However, when the Ignore Transmit Underrun bit is set (CTRLC.IGNTUR = 1), the SPI can re-sample the underrun condition and continue to transmit data at each frame boundary. If the SPI is a frame host, new data written to the DATA buffer during a frame when an underrun condition exists does not get transmitted during that frame. But, the SPI evaluates the DATA buffer continuously after the last frame. If data is in it, the SPI generates a frame sync and transmits the data. If the SPI is a frame client, its transmit logic evaluates DATA buffer for underrun during the next FSYNC. If the buffer contains data when a valid FSYNC is detected, SPI transmits that data. If not, the SPI transmits zero data until the end of the frame.

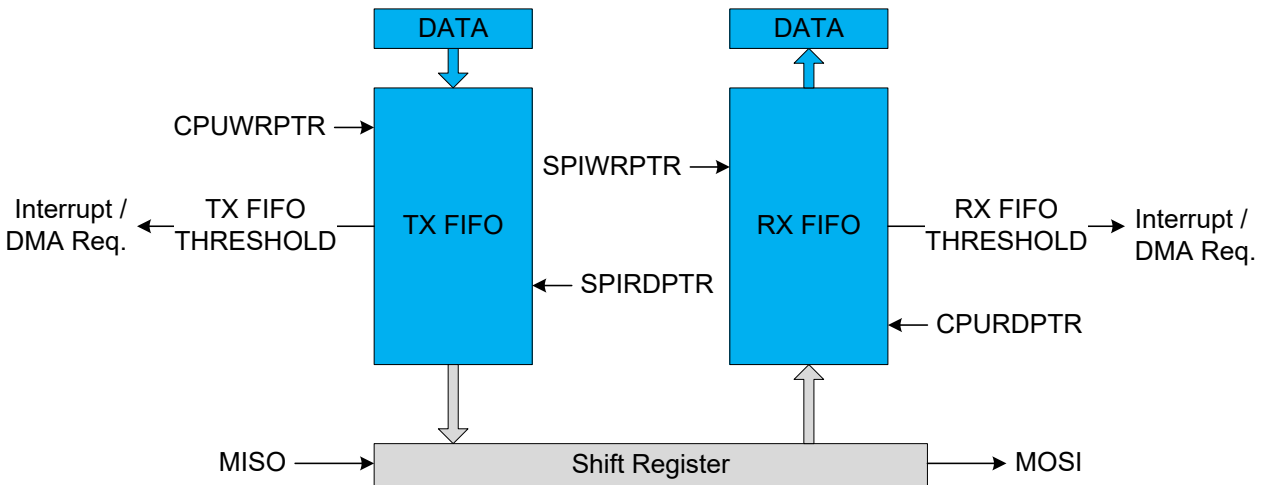
In the case of client frames, if a new FSYNC is received during an ongoing frame transaction, the transaction length error status bit is set, and the error interrupt flag is set. The module will continue to send data from the internal buffer.

34.6.4.3.10 FIFO Operation

The SPI embeds up to 16-bytes FIFO capability. The receive/transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in the CTRLC register is set to a '1' (CTRLC.FIFOEN = 1). By default, the FIFO can act as a 0-bit array, or as a 0-bit array, depending on the setting of the CTRLC.DATA32B bit. The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the SPI Write pointer (SPIWRPTR) and the SPI Read pointer (SPIRDPTR). All of these pointers reset to '0'. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the SPIWRPTR and SPIRDPTR are native to the SPI domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR becomes the current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the

SPIWRPTR / SPIRDPTR is logically the current shift register. Physically, the receive (shift-in) portion of the shift register is a single register located in the SCK clock domain, and the transmit (shift-out) portion is the buffer location pointed to by the SPIRDPTR. When a full word / byte is clocked into the SPI shift register, it is copied into the location pointed by SPIWRPTR.

Figure 34-30. FIFO Overview



The interrupts and DMA triggers are generated according to FIFO threshold settings in the Control C register (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD).

The Data Register Empty interrupt flag, and the DMA TX trigger respectively, are generated when the available place in the TX FIFO is equal or higher than the threshold value defined by the CTRLC.TXTRHOLD settings. The Transfer complete interrupt is generated when the TX FIFO is empty and the last bit is shifted out.

The Receive Complete interrupt flag, and the DMA RX trigger respectively, are generated when the number of bytes present in the RX FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD settings. The ERROR interrupt flag is generated when both RX shifter and the RX FIFO are full.

The FIFO is fully accessible if the SERCOM is halted, by writing the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective FIFO Clear bit in the Control E register (CTRLE.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredictable results.

Pointer Operation in Host Mode

In Host mode, the transmit / receive sequence is started by the CPU writing one or more transmit words into the TX FIFO. The CPUWRPTR is incremented by 1 every time the CPU writes a word to the memory array. As soon as the CPUWRPTR becomes not-equal to SPIRDPTR (FIFOSPACE.TXSPACE != 0), the SPI transmits the data pointed to by the SPIRDPTR through MOSI. When a complete word is shifted in, the SPIRDPTR is compared to CPUWRPTR. If they are not equal, SPIRDPTR is incremented, another byte / word is shifted in/out, and so on. When the CPU completes a read from the RX FIFO location (FIFOSPACE.RXSPACE != 0), the CPURDPTR pointer is incremented. When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The module will not respond to SCK transitions while BUFOVF is '1', effectively disabling the module until software reads DATA register.

All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to '0'. CPURDPTR will not increment past SPIWRPTR. In other words, if CPURDPTR = SPIWRPTR, and the CPU attempts another read, the pointer will stay at the value of SPIWRPTR.

Pointer Operation in Client Mode

In Client mode, the transmit / receive sequence is started by the SPI receiving an SCK clock pulse. As soon as an SCK pulse is received, the SPI transmits the data pointed to by the SPIRDPTR. When a complete data is shifted in, the SPIRDPTR / SPIWRPTR are incremented, another data are shifted in/ out, and so on. The newly received data is written to the RX FIFO location pointed to by SPIWRPTR. The CPUWRPTR is incremented by one every time the CPU writes a new data to the TX FIFO memory array. If the CPUWRPTR is pointing to location n, and SPIWRPTR is pointing to location n, and a wrap is not detected, the CPUWRPTR will auto-increment to SPIWRPTR, to keep up with SPIWRPTR. This is so that the next data to be transmitted will be placed in the correct position in the storage element. The CPU can read data from RX FIFO as long as FIFOSPACE.RXSPACE != 0. When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The module will not respond to SCK transitions while BUFOVF is '1', effectively disabling the module until software reads RX FIFO. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then roll over to '0'. CPURDPTR will not increment past SPIWRPTR. In other words, if CPURDPTR = SPIWRPTR, and the CPU attempts another read, the pointer will stay at the value of SPIWRPTR.

34.6.4.3.11 Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodic service by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

This peripheral can be forced to halt operation during debugging see (DBGCTRL) register for details.

34.6.4.4 DMA Requests and Interrupts

Table 34-12. Module Request for SERCOM SPI

Condition	Request DMA	Interrupt
Standard (DRE): Data Register Empty FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty	Yes (request cleared when data is written)	Yes
Standard (RXC): Receive Complete FIFO (RXC): at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed.	Yes (request cleared when data is read)	Yes
Standard (TXC): Transmit Complete FIFO (TXC): Transmit Complete and TX FIFO is empty	N/A	Yes
Client Select low (SSL)	N/A	Yes
Error (ERROR)	N/A	Yes

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake-up the device from any Sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Slave Select Low (SSL)
- Error (ERROR)

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to Device DMAC – Direct Memory Access Controller for details.

34.6.4.5 Sleep Mode Operation

The behavior in Sleep mode depends on the host/client configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Host operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOM_CORE will continue to run in Idle Sleep mode and in Standby Sleep mode. Any interrupt can wake-up the device.
- Host operation, CTRLA.RUNSTDBY=0: GLK_SERCOM_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Client operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake-up the device
- Client operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

34.6.4.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").

The following bits are synchronized when written:

- The Software Reset bit in the CTRLA register (CTRLA.SWRST)
- The Enable bit in the CTRLA register (CTRLA.ENABLE)
- The Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the CTRLB register (CTRLB.TXEN)

Note: CTRLB.RXEN is write-synchronized somewhat differently than other registers. See the CTRLB register for details.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description. If a write-synchronized register is written while a synchronization is ongoing, a APB (Advanced Peripheral Bus) Error will be generated.

34.7 SERCOM I²C

34.7.1 Overview

The Inter-Integrated Circuit (I²C) interface is one of the available modes in the Serial Communication Interface module (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in the Block Diagram. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM.

A SERCOM instance can be configured to be either an I²C host or an I²C client. Both host and client have an interface containing a Shift register, a transmit buffer and a receive buffer. The I²C host mode uses the SERCOM baud-rate generator, while the I²C client mode uses the SERCOM address match logic.

In addition, the I²C Host uses the SERCOM baud-rate generator, while the I²C Client uses the SERCOM address match logic .

34.7.2 Features

SERCOM I²C includes the following features:

- Host or Client Operation
- Can be used with DMA
- Philips I²C Compatible
- SMBus Compatible
- PMBus™ Compatible

- Support of 100 kHz and 400 kHz, 1 MHz I²C mode
- 32-bit Data Extension for better system bus utilization
- 16-bytes internal FIFO
- 4-Wire Operation Supported
- Physical interface includes:
 - Slew-rate limited outputs
 - Filtered inputs
- Client Operation:
 - Operation in all Sleep modes
 - Wake-up on address match
 - 7-bit and 10-bit Address match in hardware for:
 - Unique address and/or 7-bit general call address
 - Address range
 - Two unique addresses can be used with DMA

Table 34-13. SERCOM I²C Features Depending on Package

Feature	SERCOM I ² C Features						
	0		1		2 / 3		4 / 5
I ² C speed	HS		FM+	HS	FM+		-
IOSET	1	2	1 / 2	3	1	2	-
5V tolerant support	-	-	Yes	-	-	-	-
secure capable	-	-	Yes	-	-	-	-
available on 48 pins	x	-	x	-	x	-	-
available on 64 pins	x	-	x	-	x	-	-
available on 100 pins	x	x	x	x	x	x	-

34.7.3 Block Diagram

Figure 34-31. I²C Single-Host Single-Client Interconnection

34.7.4 Clocks

The SERCOM bus clock (CLK_SERCOM_APB) can be enabled and disabled in the Main Clock Controller. Refer to the Peripheral Clock Masking for details and default status of this clock.

Two generic clocks are used by SERCOM, GCLK_SERCOMx_CORE and CLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I²C when working as a host. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These two clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the I²C.

Note: The "GCLK_SERCOMn_SLOW" clock is ONLY used for I2C mode exclusively.

These generic clocks are asynchronous to the bus clock (CLK_SERCOM_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

34.7.5 Functional Description

34.7.5.1 Principle of Operation

The I²C interface uses two physical lines for communication:

- Serial Data Line (SDA) for data transfer

- Serial Clock Line (SCL) for the bus clock

A transaction starts with the I²C host sending the Start condition, followed by a 7-bit address and a direction bit (read or write to/from the client).

The addressed I²C client will then Acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is Not Acknowledged (NACK), whether by the I²C client or host, the I²C host takes action by either terminating the transaction by sending the Stop condition, or by sending a repeated start to transfer more data.

The following figure illustrates the possible transaction formats. These symbols will be used in the following descriptions.

Figure 34-32. Transaction Diagram Symbols

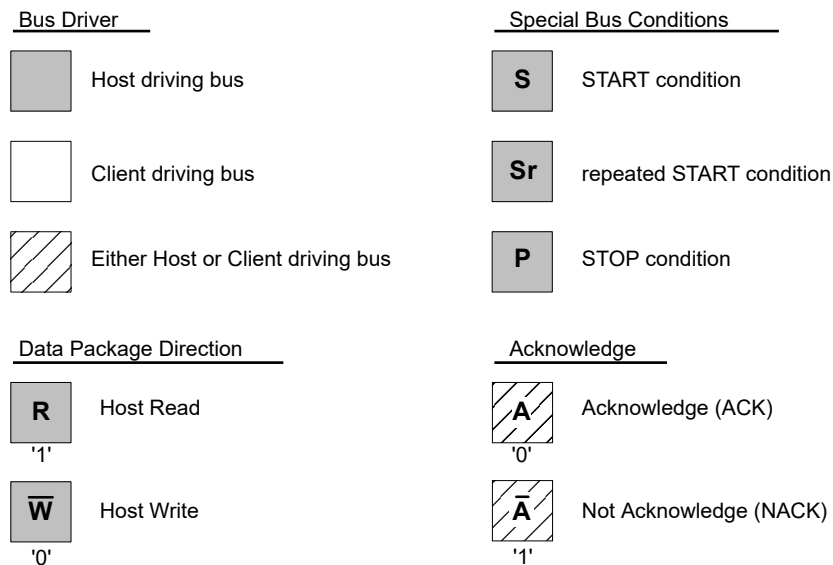
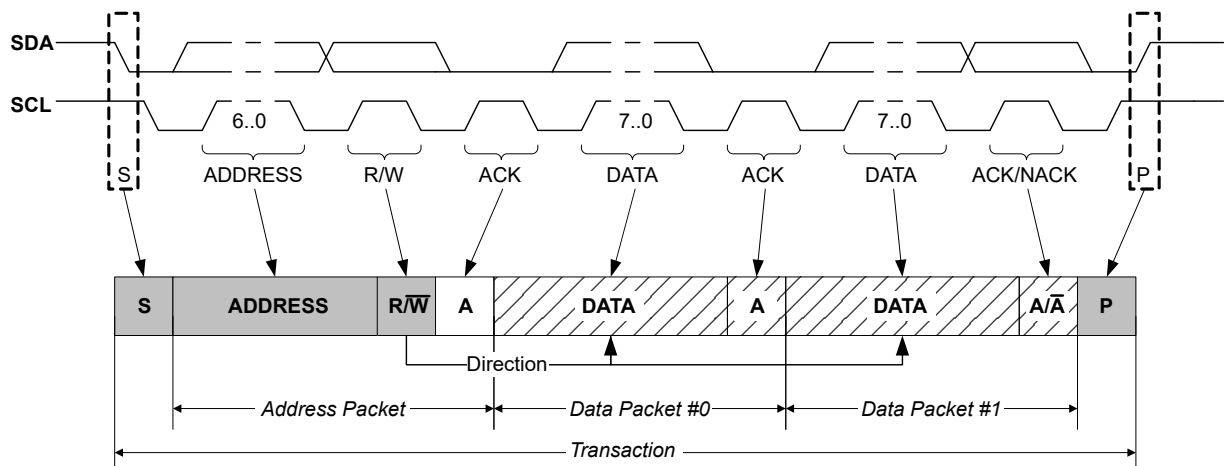


Figure 34-33. Basic I²C Transaction Diagram



34.7.5.2 Basic Operation

34.7.5.2.1 Initialization

The following registers are enable-protected, meaning they can be written only when the I2C interface is disabled (CTRLA.ENABLE is '0'):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control E register (CTRLE), except Acknowledge Action (CTRLE.ACKACT) and Command (CTRLE.CMD) bits
- Control D register (CTRLD) Baud register (BAUD) Address register (ADDR) in Client operation.

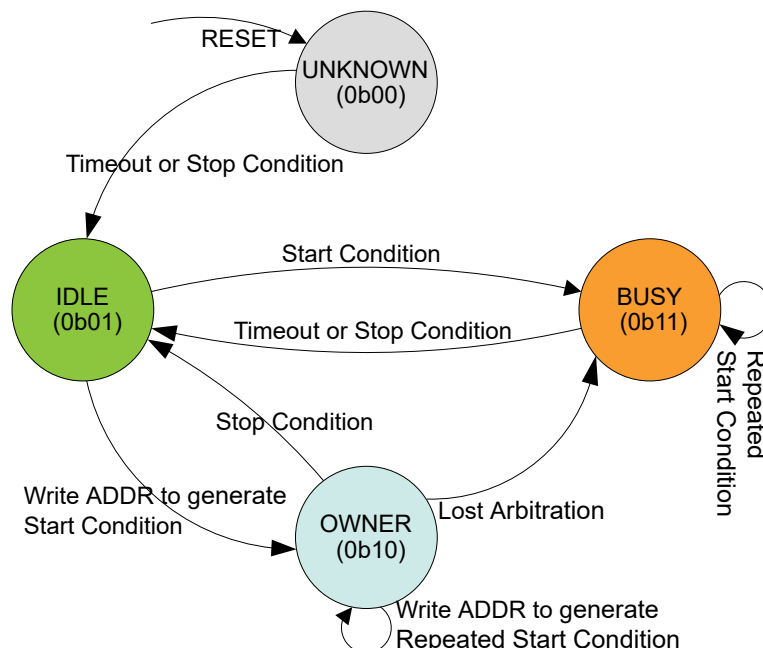
When the I2C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I2C is being disabled, writing to these registers will be completed after the disabling. Enable-protection is denoted by the "Enable-Protection" property in the register description

1. Select I²C Host or Client mode by writing 0x4 (Client mode) or 0x5 (Host mode) to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
 2. If desired, select the SDA Hold Time value in the CTRLA register (CTRLA.SDAHOLD).
 3. In Client mode, the minimum client setup time for the SDA can be selected in the SDA Setup Time bit group in the Control C register (CTRLC.SDASETUP).
 4. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
 5. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUT).
 6. In Host mode:
 - a. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
 - b. Write the Baud Rate register (BAUD) to generate the desired baud rate.
- In Client mode:
- a. Configure the address match configuration by writing the Address Mode value in the CTRLB register (CTRLB.AMODE).
 - b. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

34.7.5.2.2 I²C Bus State Logic

The Bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all Sleep modes with running GCLK_SERCOM_x clocks. The start and stop detectors and the bit counter are all essential in the process of determining the current Bus state. The Bus state is determined according to the following figure. Software can get the current Bus state by reading the Host Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the figure is shown in binary.

Figure 34-34. Bus State Diagram



The Bus state machine is active when the I²C host is enabled.

After the I²C host has been enabled, the Bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by writing 0b01 to STATUS.BUSSTATE
- A Stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs.

Note: Once a known Bus state is established, the Bus state logic will not re-enter the UNKNOWN state.

When the bus is IDLE it is ready for a new transaction. If a Start condition is issued on the bus by another I²C host in a multi-host setup, the bus becomes BUSY (0b11). The bus will re-enter IDLE either when a Stop condition is detected, or when a time-out occurs (inactive bus time-out needs to be configured).

If a Start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, i.e., arbitration was not lost, the I²C host can issue a Stop condition, which will change the Bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the Bus state becomes BUSY until a Stop condition is detected. A repeated Start condition will change the Bus state only if arbitration is lost while issuing a repeated start.

Note: Violating the protocol may cause the I²C to hang. If this happens it is possible to recover from this state by a software Reset (CTRLA.SWRST='1').

34.7.5.2.3 I²C Host Operation

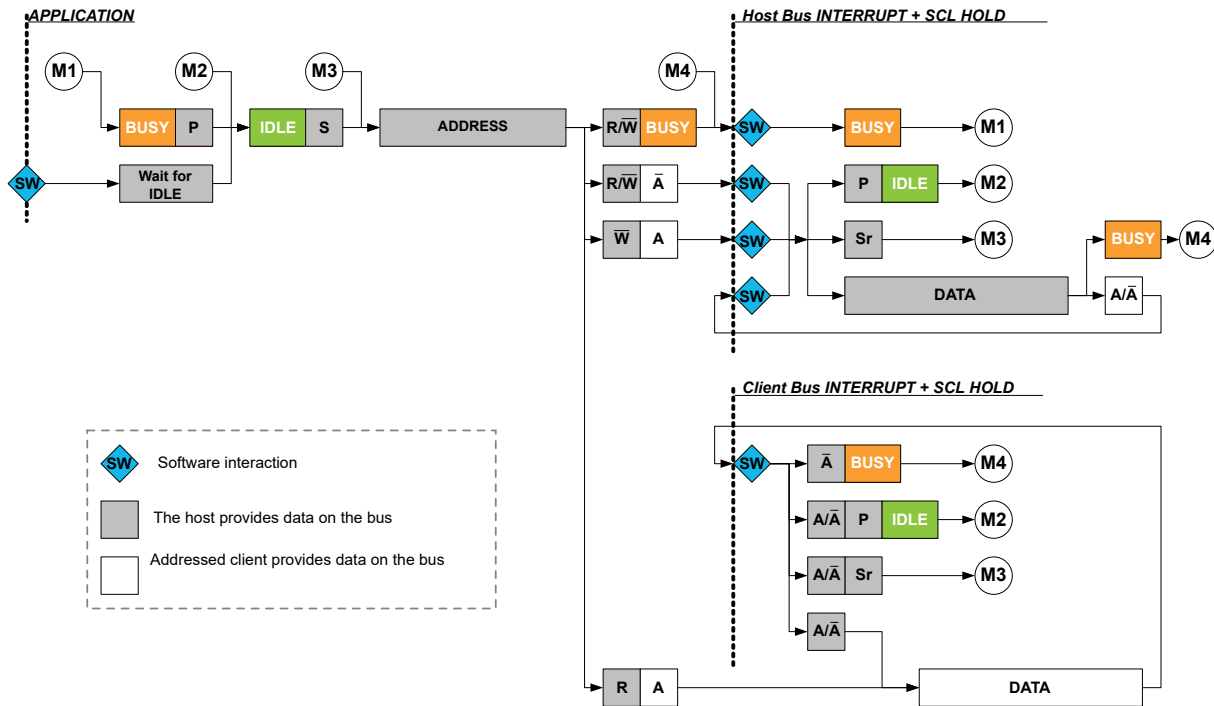
The I²C host is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations, and a Special Smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLB.SMEN).

The I²C host has two interrupt strategies.

When SCL Stretch Mode (CTRLD.SCLSM) is '0', SCL is stretched before or after the Acknowledge bit . In this mode the I²C host operates according to the following diagram. The circles labeled "Mn" (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C host operation throughout the document.

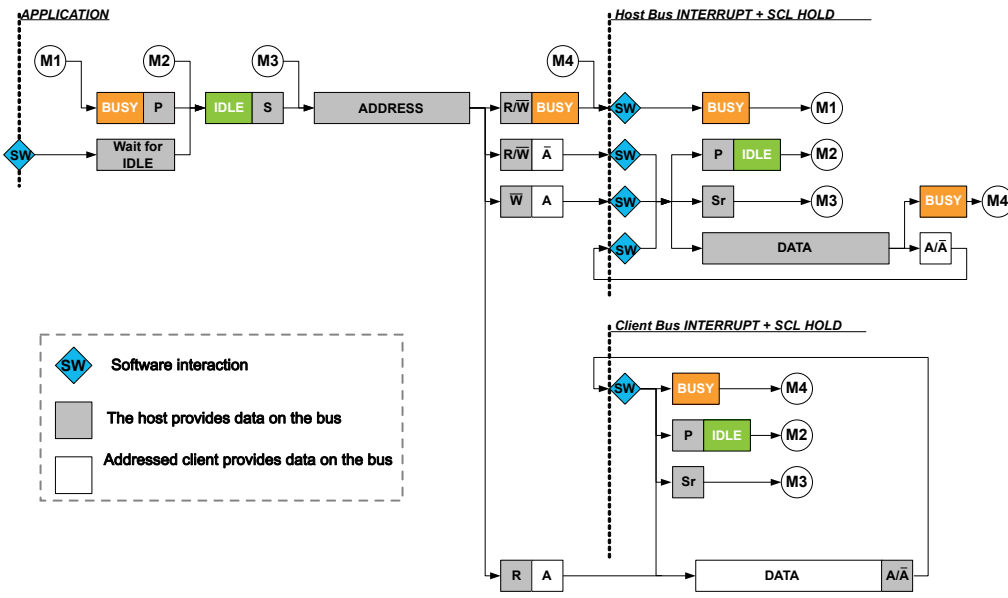
Figure 34-35. I²C Host Behavioral Diagram (SCLSM=0)



In the second strategy (CTRLD.SCLSM=1), interrupts only occur after the ACK bit, as in the following diagram. This strategy can be used when it is not necessary to check DATA before acknowledging.

Note: I²C High-speed (*Hs*) mode requires CTRLD.SCLSM=1.

Figure 34-36. I²C Host Behavioral Diagram (SCLSM=1)



Host Clock Generation

The SERCOM peripheral supports several I²C bidirectional modes:

- Standard mode (*Sm*) up to 100 kHz
- Fast mode (*Fm*) up to 400 kHz
- Fast mode Plus (*Fm+*) up to 1 MHz
- High-speed mode (*Hs*) up to 3.4 MHz

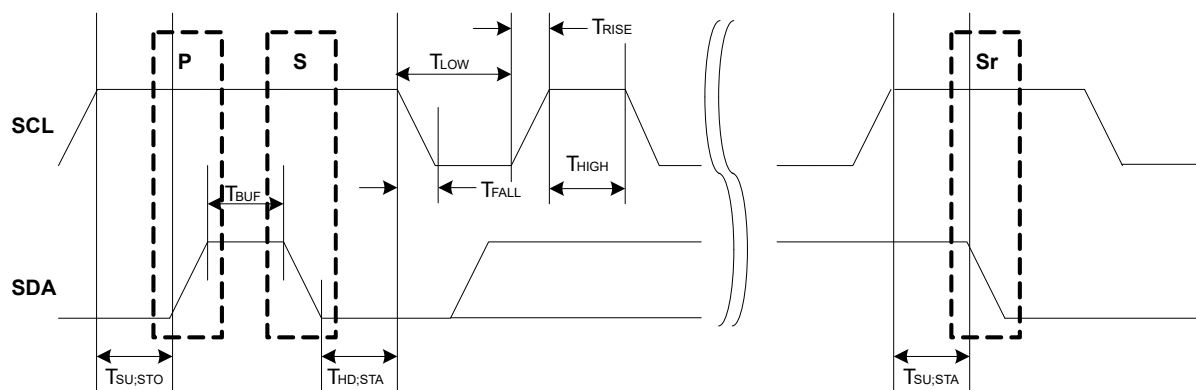
The Host clock configuration for *Sm*, *Fm*, and *Fm+* are described in *Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)*. For *Hs*, refer to *Host Clock Generation (High-Speed Mode)*.

Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm*, *Fm*, and *Fm+* mode, the Host clock (SCL) frequency is determined as described in this section:

The low (TLOW) and high (THIGH) times are determined by the Baud Rate register (BAUD), while the rise (TRISE) and fall (TFALL) times are determined by the bus topology. Because of the wired-AND logic of the bus, TFALL will be considered as part of TLOW. Likewise, TRISE will be in a state between TLOW and THIGH until a high state has been detected.

Figure 34-37. SCL Timing



The following parameters are timed using the SCL low time period T_{LOW}. This comes from the Host Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, the Host Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- **T_{LOW}**: Low period of SCL clock
- **TSU;STO**: Set-up time for stop condition
- **TBUF**: Bus free time between stop and start conditions
- **THD;STA**: Hold time (repeated) start condition
- **TSU;STA**: Set-up time for repeated start condition
- **THIGH**: Timed using the SCL high time count from BAUD.BAUD
- **TRISE**: Determined by the bus impedance; for internal pull-ups
- **TFALL**: Determined by the open-drain current limit and bus impedance; can typically be regarded as zero

The SCL frequency is given by:

$$f_{SCL} = \frac{1}{T_{LOW} + T_{HIGH} + T_{RISE}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + 2BAUD + f_{GCLK} \cdot T_{RISE}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + BAUD + BAUDLOW + f_{GCLK} \cdot T_{RISE}}$$

The following formulas can determine the SCL T_{LOW} and T_{HIGH} times:

$$T_{LOW} = \frac{BAUDLOW + 5}{f_{GCLK}}$$

$$T_{HIGH} = \frac{BAUD + 5}{f_{GCLK}}$$

Note: The I²C standard Fm+ (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.

Startup Timing The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.

Note: When timing is controlled by user, the Smart Mode cannot be enabled.

Host Clock Generation (High-Speed Mode)

For I²C Hs transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency (f_{GCLK}) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HSBAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{2 + HSBAUD + HSBAUDLOW}$$

Note: The I²C standard *Hs*(High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

Transmitting Address Packets

The I²C host starts a bus transaction by writing the I²C client address to ADDR.ADDR and the direction bit, as described in 5.6.1 Principle of Operation. If the bus is busy, the I²C host will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C host will issue a start condition on the bus. The I²C host will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C host, one of four cases will arise according to arbitration and transfer direction.

Case 1: Arbitration lost or bus error during address packet transmission

If arbitration was lost during transmission of the address packet, the Host on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C host is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the Arbitration Lost condition. In this case, the MB Interrupt flag and Host Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Host Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the Interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

Case 2: Address packet transmit complete – No ACK received

If there is no I²C client device responding to the address packet, then the INTFLAG.MB Interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I²C client is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a Stop condition (recommended) or resending the address packet by a repeated Start condition. When using SMBus logic, the client must ACK the address. If there is no response, it means that the client is not available on the bus.

Case 3: Address packet transmit complete – Write packet, Host on Bus set

If the I²C host receives an ACK response from the I²C client, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA
- Transmit a new address packet by writing ADDR.ADDR. A repeated Start condition will automatically be inserted before the address packet.
- Issue a Stop condition, consequently terminating the transaction

Case 4: Address packet transmit complete – Read packet, Client on Bus set

If the I²C host receives an ACK from the I²C client, the I²C host proceeds to receive the next byte of data from the I²C client. When the first data byte is received, the Client on Bus bit in the Interrupt

Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Let the I²C host continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in Smart mode.
- Transmit a new address packet
- Terminate the transaction by issuing a Stop condition

Note: An ACK or NACK will be automatically transmitted if Smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

Transmitting Data Packets

When an address packet with direction Host Write is transmitted successfully, INTFLAG.MB will be set. The I²C host will start transmitting data via the I²C bus by writing to DATA.DATA, and monitor continuously for packet collisions.

If a collision is detected, the I²C host will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I²C host will receive an ACK bit from the I²C client, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I²C Host on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I²C host is not allowed to continue transmitting data packets if a NACK is received from the I²C client.

Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I²C host will already have received one data packet. The I²C host must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I²C host will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the Smart mode.

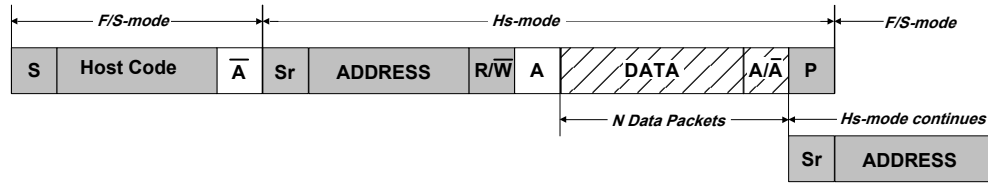
High-Speed Mode

High-speed transfers are a multi-step process, for further information, refer to the following figure.

First, a host code (0b00001nnn, where 'nnn' is a unique host code) is transmitted in Full-speed mode, followed by a NACK since no client should acknowledge. Arbitration is performed only during the Full-speed Host Code phase. The host code is transmitted by writing the host code to the Address register (ADDR.ADDR) and writing the High-speed bit (ADDR.HS) to '0'.

After the host code and NACK have been transmitted, the host write interrupt will be asserted. In the meantime, the client address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the host will generate a repeated start, followed by the client address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

Figure 34-38. High Speed Transfer



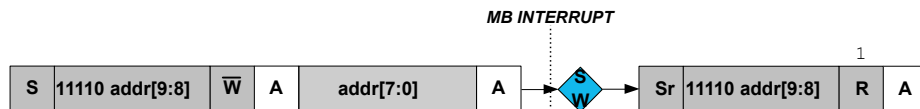
Transmitting in High-speed mode requires the I²C host to be configured in High-speed mode (CTRLD.SPEED=0x2) and the SCL Clock Stretch mode (CTRLD.SCLSM) bit set to '1'.

10-bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see 10-bit Address Transmission for a Read Transaction. The addressed client acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the host must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the host receives a NACK after the first byte, the Write Interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more clients, then the host will proceed to transmit the second address byte and the host will first see the Write Interrupt flag after the second byte is transmitted. If the transaction direction is read-from-client, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

Figure 34-39. 10-bit Address Transmission for a Read Transaction



This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. Once the Host on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 address[9:8]1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

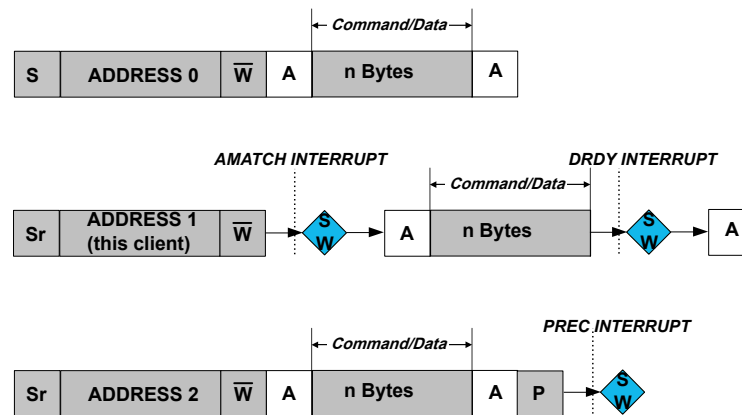
PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set if the client has been addressed since the last STOP condition. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the clients addressed during the group command, they all begin executing the command they received.

The following figure shows an example where this client, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple clients addressed before and after this client. Eventually, at the end of the group command, a single STOP is generated by the host. At this point a STOP interrupt is asserted.

Figure 34-40. PMBus Group Command Example



34.7.5.2.4 I²C Client Operation

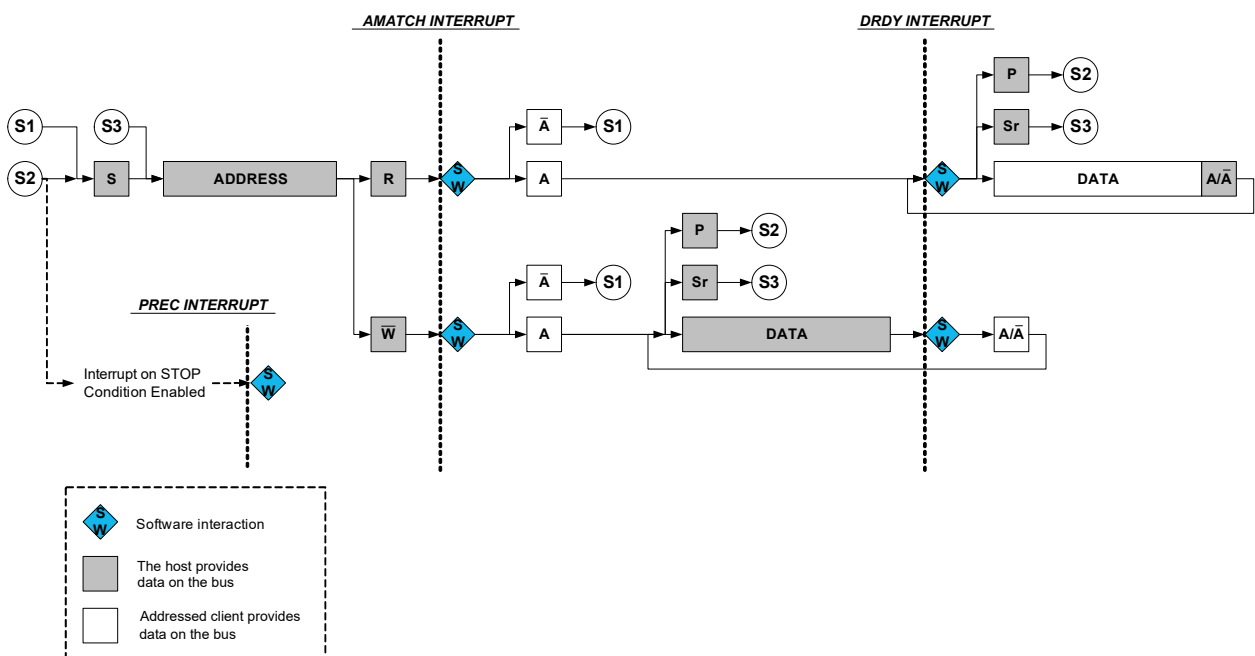
The I²C client is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control D register (CTRLD.SMEN).

The I²C client has two interrupt strategies.

When SCL Stretch Mode bit (CTRLD.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C client operates according to the following figure. The circles labeled "Sn" (S1, S2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C client operation throughout the document.

Figure 34-41. I²C Behavioral Diagram (SCLSM=0)

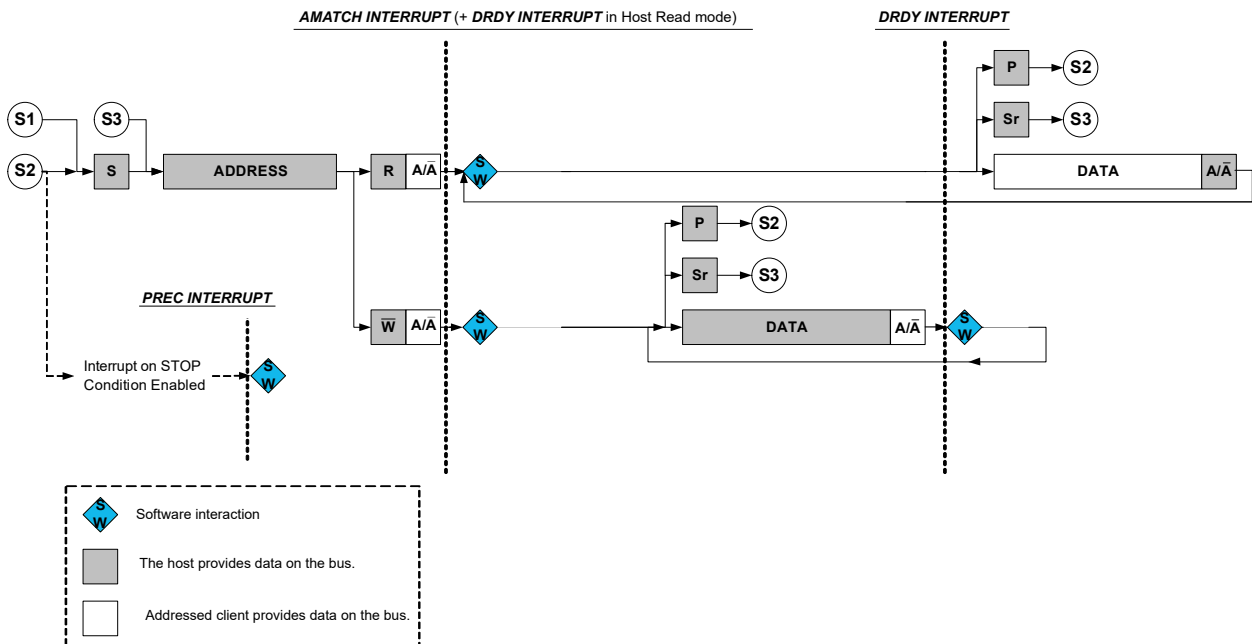


In the second strategy (CTRLD.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Client Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to

check DATA before acknowledging. For host reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for host writes, the first data interrupt will be seen after the first data byte has been received by the client and the acknowledge bit has been sent to the host.

Note: For I²C High-speed mode (*Hs*), SCLSM=1 is required.

Figure 34-42. I²C Client Behavioral Diagram (SCLSM=1)



Receiving Address Packets (SCLSM = 0)

When CTRLD.SCLSM=0, the I²C client stretches the SCL line according to the I²C Behavioral Diagram (SCLSM0). When the I²C client is properly configured, it will wait for a Start condition.

When a Start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C client will wait for a new Start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C host, one of two cases will arise based on transfer direction.

Case 1: Address packet accepted – Read flag set

The STATUS.DIR bit is '1', indicating an I²C host read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C client hardware will set the Data Ready bit in the Interrupt Flag

register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C client will wait for a new Start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I²C client Command bit field in the Control E register (CTRL.E.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRL.E.ACKACT bit.

Case 2: Address packet accepted – Write flag set

The STATUS.DIR bit is cleared, indicating an I²C host write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C client will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C client will wait for a new Start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I²C client command CTRL.E.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRL.E.ACKACT bit.

Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I²C client will stretch the SCL line only after an ACK, see Client Behavioral Diagram(SCLSM=1). When the I²C client is properly configured, it will wait for a Start condition to be detected. When a Start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I²C client will wait for a new Start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control E register (CTRL.E.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C host, INTFLAG.AMATCH be set to '1' to clear it.

Receiving and Transmitting Data Packets

After the I²C client has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA.

When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I²C client will send an acknowledge according to CTRL.E.ACKACT.

Case 1: Data received

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

Case 2: Data sent

When a byte transmission is successfully completed, the INTFLAG.DRDY Interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I²C client must expect a stop or a repeated start to be received. The I²C client must release the data line to allow the I²C host to generate a stop or repeated start. Upon detecting a Stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I²C client will return to IDLE state.

High-Speed Mode

When the I²C client is configured in High-speed mode (*Hs*, CTRLD.SPEED=0x2) and CTRLD.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the client recognizes a START followed by a host code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The client will then remain in High-speed mode until a STOP is received.

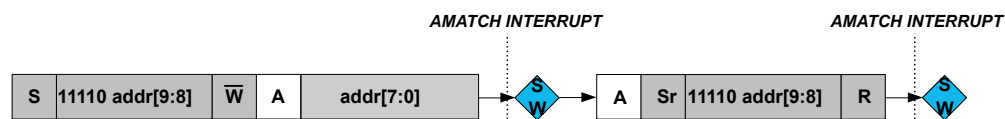
10-bit Addressing (Client)

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit client address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address Interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of 11110 ADDR[9:8] 1, and the second address interrupt will be received with the DIR bit set. The client matches on the second address as it was addressed by the previous 10-bit address.

Figure 34-43. 10-bit Addressing



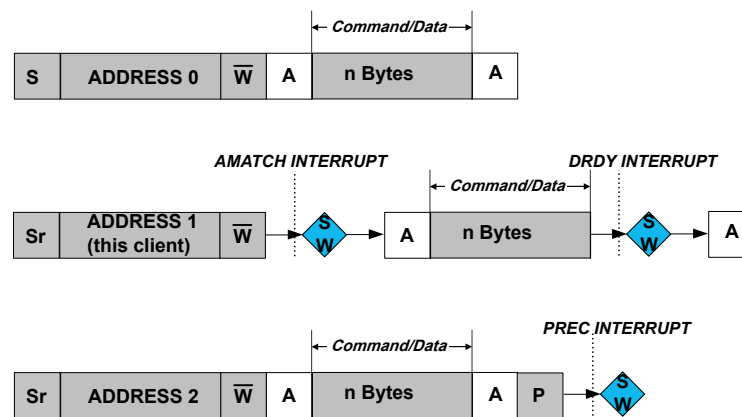
PMBus Group Command

When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set if the client has been addressed since the last STOP condition. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the clients addressed during the group command, they all begin executing the command they received.

The following figure shows an example where this client, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple clients addressed before and after this client. Eventually, at the end of the group command, a single STOP is generated by the host. At this point a STOP interrupt is asserted.

Figure 34-44. PMBus Group Command Example



34.7.5.3 Additional Features

34.7.5.3.1 SMBus

The I²C includes three hardware SCL low time-outs, which allow a time-out to occur for SMBus SCL low time-out, host extend time-out, and client extend time-out. This allows for SMBus functionality. These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32 KHz oscillator. The I²C interface also allows for a SMBus compatible SDA hold time.

- **T_{TIMEOUT}**: SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLD.LOWTOUTEN.
- **T_{LOW:SEXT}**: Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a client device in a single message from the initial START to the STOP. It is enabled by CTRLD.SEXTTOEN.
- **T_{LOW:MEXT}**: Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the host device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLD.MEXTTOEN.

The SMBus-compatible logic levels are enabled by writing the SMBus Input Buffer Enable bit in Control A register (CTRLD.SMBUFEN).

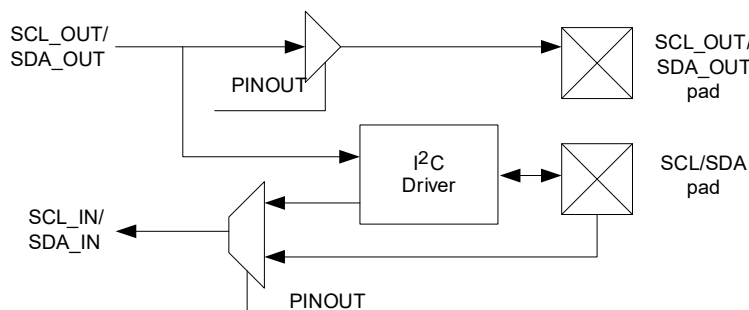
34.7.5.3.2 Smart Mode

The I²C interface has a Smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I²C protocol. The Smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRL.E.ACKACT) as soon as DATA.DATA is read.

34.7.5.3.3 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control D register (CTRLD.PINOUT) will enable 4-Wire mode operation. In this mode, the internal I²C tri-state drivers are bypassed, and an external I²C compliant tri-state driver is needed when connecting to an I²C bus.

Figure 34-45. I²C Pad Interface



34.7.5.3.4 Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding Interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the client acknowledges the address. At this point, the software can either issue a Stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

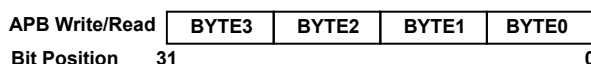
34.7.5.3.5 32-bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B=1). When enabled, write and read transaction to/from the DATA register are 32 bit in size.

If frames are not multiples of 4 Bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins.

The following figure shows the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0 to 3.

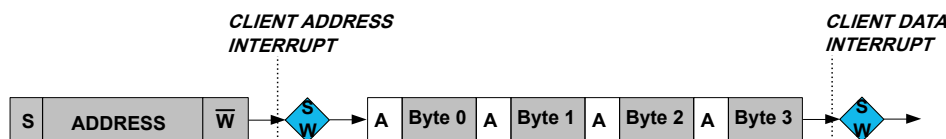
Figure 34-46. 32-bit Extension Byte Ordering



32-bit Extension Client Operation

The figure below shows a transaction with 32-bit Extension enabled (CTRLC.DATA32B=1). In client operation, the Address Match interrupt in the Interrupt Flag Status and Clear register (INTFLAG.AMATCH) is set after the address is received and available in the DATA register. The Data Ready interrupt (INTFLAG.DRDY) will then be raised for every 4 Bytes transferred.

Figure 34-47. 32-bit Extension Client Operation



The LENGTH register can be written before the frame begins, or when the AMATCH interrupt is set. If the frame size is not LENGTH.LEN Bytes, the Length Error status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 4 Bytes, the final INTFLAG.DRDY interrupt is raised when the last Byte is received for host reads. For host writes, the last data byte will be automatically NACKED. On address recognition, the internal length counter is reset in preparation for the incoming frame.

High Speed transactions start with a Full Speed Host Code. When a Host Code is detected, no data is received and the next expected operation is a repeated start. For this reason, the length is not

counted after a Host Code is received. In this case, no Length Error (STATUS.LENERR) is registered, regardless of the LENGTH.LENEN setting.

When SCL clock stretch mode is selected (CTRLD.SCLSM=1) and the transaction is a host write, the selected Acknowledge Action (CTRLE.ACKACT) will only be used to ACK/NACK each 4th byte. All other bytes are ACKed. This allows the user to write CTRLE.ACKACT=1 in the final interrupt, so that the last byte in a 32-bit word will be NACKed.

Writing to the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 Bytes, the remainder will be lost.

32-bit Extension Host Operation

When using the I²C configured as Host, the Address register must be written with the desired address (ADDR.ADDR), and optionally, the transaction Length and transaction Length Enable bits (ADDR.LEN and ADDR.LENEN) can be written. When ADDR.LENEN is written to '1' along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. Then, the ADDR.LEN bytes are transferred, followed by an automatically generated NACK (for host reads) and a STOP.

The INTFLAG.SB or INTFLAG.MB are raised for every 4 Bytes transferred. If the transaction is a host read and ADDR.LEN is not a multiple of 4 Bytes, the final INTFLAG.SB is set when the last byte is received.

When SCL clock stretch mode is enabled (CTRLD.SCLSM=1) and the transaction is a host read, the selected Acknowledge Action (CTRLE.ACKACT) will only be used to ACK/NACK each 4th Byte. All other bytes are ACKed. This allows the user to set CTRLE.ACKACT=1 in the final interrupt, so that the last byte in a 32-bit word will be NACKed.

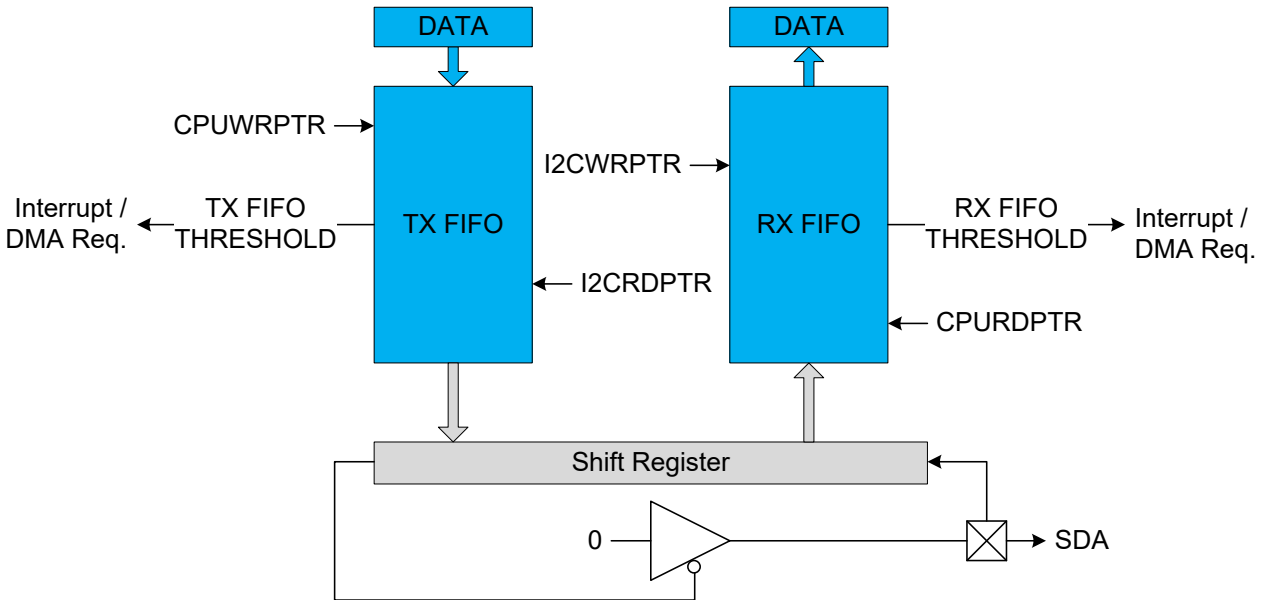
If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP will be automatically generated, and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

34.7.5.3.6 FIFO Operation

For better system bus utilization, the I²C embeds up to 16-bytes FIFO capability. The receive / transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in CTRL register is set (CTRL.FIFOEN = 1). By default, the FIFO can act as a 16-by-8-bit array, or as a 4-by-32-bit array, depending on the setting of the CTRL.DATA32B bit.

The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the I²C Write pointer (I2CWRPTR) and the I²C Read pointer (I2CRDPTR). All of these pointers reset to '0'. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the I2CWRPTR and I2CRDPTR are native to the I²C domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR becomes the current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the I2CWRPTR / I2CRDPTR is logically the current shift register.

Figure 34-48. FIFO Overview



When using the I²C configured as Host, the Address register must be written with the desired address (ADDR.ADDR), and optionally, the transaction Length and transaction Length Enable bits (ADDR.LEN and ADDR.LENEN) can be written if the 32-bit extension is enabled (CTRLC.DATA32B).

In client operation, the Address Match interrupt in the Interrupt Flag Status and Clear register (INTFLAG.AMATCH) is set after the address is received, and the SCL clock is stretched as long as the FIFO is empty in host read mode.

The FIFO threshold settings allow (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD) allow flexible interrupt, DMA trigger and bus condition generations, as described below.

The FIFO is fully accessible if the SERCOM is halted, by writing the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective FIFO Clear bit in the Control B register (CTRLB.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredictable results.

Hardware Actions in Host Mode

Table 34-14. Interrupts Request Conditions for Valid SERCOM I²C Host Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Action
Host Write	0	0	0	INTFLAG.TXFE = 1 if TX FIFO is empty
	0	1	0	- INTFLAG.TXFE = 1 if TX FIFO threshold reached - INTFLAG.MB = 1 if TX FIFO is empty and SCL hold
	0	1	1	- INTFLAG.TXFE = 1 if TX FIFO is empty - INTFLAG.TXFE = 1 if TX FIFO threshold is reached - INTFLAG.MB = 1 if TX FIFO is empty and SCL hold, or length transaction is completed
	1	0	0	INTFLAG.TXFE = 1 if TX FIFO is empty
	1	1	0	INTFLAG.TXFE = 1 if TX FIFO threshold reached
	1	1	1	INTFLAG.MB = 1 if TX FIFO is empty and SCL hold, or length transaction is completed

Table 34-14. Interrupts Request Conditions for Valid SERCOM I²C Host Configurations (continued)

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Action
Host Read	0	0	0	- INTFLAG.SB = 1 if RX FIFO is full - INTFLAG.RXFE = 1 if RX FIFO threshold reached or length transaction is completed
	0	1	0	
	0	1	1	
	1	0	0	
	1	1	0	
	1	1	1	

Table 34-15. Bus Actions for Valid SERCOM I²C Host Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Actions
Host Write	0	0	0	SCL hold if TX FIFO is empty
	0	1	0	
	0	1	1	- SCL hold if TX FIFO is empty and length transaction not completed - Issue STOP when transaction is completed
	1	0	0	- SCL hold if TX FIFO is empty, when no automatic stop is sent - STOP is sent on SW decision
	1	1	0	- SCL hold if TX FIFO is empty, when no automatic stop is sent - STOP is sent on SW decision
	1	1	1	- SCL hold if TX FIFO is empty and length transaction is not completed - Issue STOP when transaction is completed
Host Read	0	0	0	SCL hold if RX FIFO is full
	0	1	0	SCL hold if RX FIFO is full
	0	1	1	- SCL stretched if RX FIFO is full - ACK/NACK last frame byte, depending on Acknowledge Action (CTRLB.ACKACT) - ACK all other bytes
	1	0	0	SCL stretched if RX FIFO is full
	1	1	0	
	1	1	1	- SCL stretched if RX FIFO is full - ACK/NACK last frame byte, depending on Acknowledge Action (CTRLB.ACKACT) - ACK all other bytes

Hardware Actions in Client Mode

Table 34-16. Interrupt Request Conditions for Valid SERCOM I²C Client Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Condition
Host Write	0	0	0	- INTFLAG.DRDY = 1 if RX FIFO is full - INTFLAG.RXFF = 1 RX FIFO threshold is reached or length transaction is completed
	0	1	0	
	0	1	1	
	1	0	0	
	1	1	0	
	1	1	1	
Host Read	0	0	0	- INTFLAG.DRDY = 1 if TX FIFO is empty and SCL hold - INTFLAG.TXFE = 1 if TX FIFO is empty or TX FIFO threshold is reached
	0	1	0	
	0	1	1	
	1	0	0	
	1	1	0	
	1	1	1	

Table 34-17. Bus Actions for Valid SERCOM I²C Client Configurations

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Actions
Host Write	0	0	0	- Byte mode operation - SCL stretched if RX FIFO is full
	0	1	0	32-bit mode operation - SCL stretched if RX FIFO is full - ACK/NACK each 4th byte, depending on Acknowledge Action (CTRLB.ACKACT) - ACK all other bytes
	0	1	1	32-bit mode operation with length control - SCL stretched if RX FIFO is full - ACK/NACK last byte of the frame, depending on Acknowledge Action (CTRLB.ACKACT) - ACK all other bytes
	1	0	0	- SCL stretched if RX FIFO is full - ACK all bytes received
	1	1	0	32-bit mode operation - SCL stretched if RX FIFO is full - ACK/NACK each 4th byte, depending on Acknowledge Action (CTRLB.ACKACT) - ACK all other bytes
	1	1	1	32-bit mode operation with length control - SCL stretched if RX FIFO is full - ACK/NACK last byte of the frame, depending on Acknowledge Action (CTRLB.ACKACT) - ACK all other bytes

Table 34-17. Bus Actions for Valid SERCOM I²C Client Configurations (continued)

Direction	CTRLB.SMEN	CTRLC.DATA32B	LENGTH.LENEN	Actions
Host Read	0	0	0	• SCL stretched if TX FIFO is empty
	0	1	0	• SCL stretched if TX FIFO is empty
	0	1	1	• SCL stretched if TX FIFO is empty and length transaction is not completed
	1	0	0	• SCL stretched if TX FIFO is empty
	1	1	0	• SCL stretched if TX FIFO is empty
	1	1	1	• SCL stretched if TX FIFO is empty and length transaction is not completed

34.7.5.4 DMA and Interrupts

This chapter provides DMA and interrupt conditions when the FIFO is disabled. For details when the FIFO is enabled, refer to FIFO Support.

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the Interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See the INTFLAG (Client) or INTFLAG (Host) register for details on how to clear Interrupt flags.

Table 34-18. Module Request for SERCOM I²C Client

Condition	Request DMA	Interrupt	Event
Data needed for transmit (TX) (Client Transmit mode)	Yes (request cleared when data is written)		N/A
Data received (RX) (Client Receive mode)	Yes (request cleared when data is read)		
Data Ready (DRDY)		Yes	
Address Match (AMATCH)		Yes	
Stop received (PREC)		Yes	
TX FIFO Empty (TXFE)		Yes	
RX FIFO Full (RXFF)		Yes	
Error (ERROR)		Yes	

Table 34-19. Module Request for SERCOM I²C Host

Condition	Request DMA	Interrupt	Event
Data needed for transmit (TX) (Host Transmit mode)	Yes (request cleared when data is written)		N/A
Data needed for transmit (RX) (Host Transmit mode)	Yes (request cleared when data is read)		
Host on Bus (MB)		Yes	
Stop received (SB)		Yes	
TX FIFO Empty (TXFE)		Yes	
RX FIFO Full (RXFF)		Yes	
Error (ERROR)		Yes	

34.7.5.4.1 DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

Client DMA

When using the I²C client with DMA, an address match will cause the address Interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I²C client generates the following requests:

- Write data received (RX): If the FIFO is disabled, the request is set when host write data is received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached (CTRLC.RXTRHOLD). The request is cleared when DATA is read.
- Read data needed for transmit (TX): If the FIFO is disabled, the request is set when data is needed for a host read operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

Host DMA

When using the I²C host with DMA, the ADDR register must be written with the desired address (ADDR.ADDR) and transaction length (ADDR.LEN), with transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to 1 along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. DMA is then used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for host reads) and a STOP.

If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I²C host generates the following requests:

- Read data received (RX): If the FIFO is disabled, the request is set when host read data is received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached. The request is cleared when DATA is read.
- Write data needed for transmit (TX): If the FIFO is disabled, the request is set when data is needed for a host write operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

34.7.5.4.2 Interrupts

The I²C client has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- RX FIFO Full (RXFF)
- TX FIFO Empty (TXFE)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I²C host has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- RX FIFO Full (RXFF)
- TX FIFO Empty (TXFE)
- Client on Bus (SB)
- Host on Bus (MB)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the Interrupt flag is cleared, the interrupt is disabled or the I²C is reset. See the INTFLAG register for details on how to clear Interrupt flags.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

34.7.5.5 Sleep Mode Operation

34.7.5.5.1 I²C Host Operation

The generic clock (GLK_SERCOMx_CORE) will continue to run in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in Standby Sleep mode. Any interrupt can wake-up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake-up the device.

34.7.5.5.2 I²C Client Operation

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake-up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

34.7.5.6 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Command bits in CTRLE register (CTRLE.CMD)
- FIFO Clear bits in CTRLE register (CTRLE.FIFOCLR)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in host operation

The following registers are synchronized when written:

- Data (DATA) when in host operation
- Length (LENGTH) when in client operation

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

If a write-synchronized register is written while a synchronization is ongoing, a Bus Error exception will be generated.

34.8 Peripheral Dependencies

Table 34-20. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
SERCOM0	0x44808000	60 : ERROR 61 : RXBRK 62 : DRE/ PREC/MB 63 : TXC/ AMATCH/SB 64 : RXC/DRDY 65 : RXS/SSL/ TXFE 66 : CTSIC/RXFF	APB: CLKMSK[1] MASK4	SLOW: PCHCTRL[20] CORE: PCHCTRL[21]	26 STATUS0 PERID26	4 : RX 5 : TX		PD_CORE_SW
SERCOM1	0x4480A000	67 : ERROR 68 : RXBRK 69 : DRE/ PREC/MB 70 : TXC/ AMATCH/SB 71 : RXC/DRDY 72 : RXS/SSL/ TXFE 73 : CTSIC/RXFF	APB: CLKMSK[1] MASK5	SLOW: PCHCTRL[20] CORE: PCHCTRL[22]	27 STATUS0 PERID27	6 : RX 7 : TX		PD_CORE_SW
SERCOM2	0x4480C000	74 : ERROR 75 : RXBRK 76 : DRE/ PREC/MB 77 : TXC/ AMATCH/SB 78 : RXC/DRDY 79 : RXS/SSL/ TXFE 80 : CTSIC/RXFF	APB: CLKMSK[1] MASK6	SLOW: PCHCTRL[20] CORE: PCHCTRL[23]	28 STATUS0 PERID28	8 : RX 9 : TX		PD_CORE_SW
SERCOM3	0x4480E000	81 : ERROR 82 : RXBRK 83 : DRE/ PREC/MB 84 : TXC/ AMATCH/SB 85 : RXC/DRDY 86 : RXS/SSL/ TXFE 87 : CTSIC/RXFF	APB: CLKMSK[1] MASK7	SLOW: PCHCTRL[20] CORE: PCHCTRL[24]	29 STATUS0 PERID29	10 : RX 11 : TX		PD_CORE_SW

Table 34-20. Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
SERCOM4	0x44820000	104 : ERROR 105 : RXBRK 106 : DRE/ PREC/MB 107 : TXC/ AMATCH/SB 108 : RXC/ DRDY 109 : RXS/SSL/ TXFE 110 : CTSIC/ RXFF	APB: CLKMSK[1] MASK12	PCHCTRL[28]	34 STATUS1 PERID2	24 : RX 25 : TX		PD_CORE_SW
SERCOM5	0x44822000	111 : ERROR 112 : RXBRK 113 : DRE/ PREC/MB 114 : TXC/ AMATCH/SB 115 : RXC/ DRDY 116 : RXS/SSL/ TXFE 117 : CTSIC/ RXFF	APB: CLKMSK[1] MASK13	PCHCTRL[29]	35 STATUS1 PERID3	26 : RX 27 : TX		PD_CORE_SW

34.9 Register Summary - USART

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		DORD	CPOL	CMODE	FORM[3:0]			
		23:16	SAMPAP[1:0]		RXPO[1:0]				TXPO[1:0]	
		15:8	SAMPAP[2:0]					RXINV	TXINV	IBON
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24							LINCMD[1:0]	
		23:16	FIFOCLR[1:0]						RXEN	TXEN
		15:8			PMODE			ENC	SFDE	COLDEN
		7:0		SBMODE				CHSIZE[2:0]		
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN		DATA32B[1:0]	
		23:16			MAXITER[2:0]				DSNACK	INACK
		15:8					HDRDLY[1:0]		BRKLEN[1:0]	
		7:0					GTIME[2:0]			
0x0C	BAUD	15:8	BAUD[15:8]							
		7:0	BAUD[7:0]							
0x0E	RXPL	7:0	RXPL[7:0]							
0x0F	Reserved									
... 0x13										
0x14	INTENCLR	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	15:8								
		7:0	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
0x20	RXERRCNT	7:0								
0x21	Reserved									
0x22	LENGTH	15:8							LENEN[1:0]	
		7:0	LEN[7:0]							
0x24	Reserved									
... 0x27										
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C	Reserved									
... 0x2F										
0x30	DBGCTRL	7:0								DBGSTOP
0x31	Reserved									
... 0x33										
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

34.9.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 34-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CMODE	FORM[3:0]			
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SAMP_A[1:0]		RXPO[1:0]				TXPO[1:0]	
Access	RW	RW	RW	RW			RW	RW
Reset	0	0	0	0			0	0
Bit	15	14	13	12	11	10	9	8
	SAMP_R[2:0]					RXINV	TXINV	IBON
Access	RW	RW	RW			RW	RW	RW
Reset	0	0	0			0	0	0
Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order

This bit selects the data order when a character is shifted out from the Data register.
This bit is not synchronized.

Value	Name	Description
0x0	MSB	MSB is transmitted first.
0x1	LSB	LSB is transmitted first.

Bit 29 – CPOL Clock Polarity

This bit selects the relationship between data output change and data input sampling in synchronous mode.
This bit is not synchronized.

CPOL	TxD Change	RxD Sample
0x0	Rising XCK edge	Falling XCK edge
0x1	Falling XCK edge	Rising XCK edge

Bit 28 – CMODE Communication Mode

This bit selects asynchronous or synchronous communication.
This bit is not synchronized.

Value	Name	Description
0x0	ASYNCR	Asynchronous communication.
0x1	SYNCR	Synchronous communication.

Bits 27:24 – FORM[3:0] Frame Format

These bits define the frame format.
These bits are not synchronized.

FORM[3:0]	Description
0x0	USART frame
0x1	USART frame with parity
0x2	LIN Hosst - Break and sync generation. See LIN Command (CTRLB.LINCMD).
0x3	Reserved
0x4	Auto-baud (LIN Client) - break detection and auto-baud.
0x5	Auto-baud - break detection and auto-baud with parity
0x6 - 0xF	Reserved

Value	Name	Description
0x0	USART	USART frame
0x1	USARTP	USART frame with parity
0x2	LINBRKGEN	LIN Master Break and Synck generation
0x4	LINBRKDET	LIN Slave break detection and auto-baud
0x5	BRKDET	Break detection and auto-baud with parity

Bits 23:22 – SAMPA[1:0] Sample Adjustment

These bits define the sample adjustment.
These bits are not synchronized.

SAMPA[1:0]	16x Over-sampling (CTRLA.SAMPR=0 or 1)	8x Over-sampling (CTRLA.SAMPR=2 or 3)
0x0	7-8-9 (Recommended)	3-4-5 (Recommended)
0x1	9-10-11	4-5-6
0x2	11-12-13	5-6-7
0x3	13-14-15	6-7-8

Bits 21:20 – RXPO[1:0] Receive Data Pinout

These bits define the receive data (RxD) pin configuration.
These bits are not synchronized.

RXPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used for data reception
0x1	PAD[1]	SERCOM PAD[1] is used for data reception
0x2	PAD[2]	SERCOM PAD[2] is used for data reception
0x3	PAD[3]	SERCOM PAD[3] is used for data reception

Value	Name	Description
0x0	PAD0	SERCOM PAD0 is used for data reception
0x1	PAD1	SERCOM PAD1 is used for data reception
0x2	PAD2	SERCOM PAD2 is used for data reception
0x3	PAD3	SERCOM PAD3 is used for data reception

Bits 17:16 – TXPO[1:0] Transmit Data Pinout

These bits define the transmit data (TxD) and XCK pin configurations.
This bit is not synchronized.

TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x0	SERCOM PAD[0]	SERCOM PAD[1]	N/A	N/A
0x1	Reserved			

TXPO (continued)				
TXPO	TxD Pin Location	XCK Pin Location (When Applicable)	RTS/TE	CTS
0x2	SERCOM PAD[0]	N/A	SERCOM PAD[2]	SERCOM PAD[3]
0x3	SERCOM PAD[0]	SERCOM PAD[1]	SERCOM PAD[2]	N/A

Value	Name	Description
0x0	PAD0	PAD[0] = TxD; PAD[1] = XCK
0x2	PAD2	PAD[0] = TxD; PAD[2] = RTS; PAD[3] = CTS
0x3	PAD3	PAD[0] = TxD; PAD[1] = XCK; PAD[2] = TE

Bits 15:13 – SAMPR[2:0] Sample

These bits select the sample rate.
These bits are not synchronized.

SAMPR[2:0]	Description
0x0	16x over-sampling using arithmetic baud rate generation.
0x1	16x over-sampling using fractional baud rate generation.
0x2	8x over-sampling using arithmetic baud rate generation.
0x3	8x over-sampling using fractional baud rate generation.
0x4	3x over-sampling using arithmetic baud rate generation.
0x5-0x7	Reserved

Value	Name	Description
0x0	ARITHM16X	16x oversampling using arithmetic baud rate generation
0x1	FRAC16X	16x oversampling using fractional baud rate generation
0x2	ARITHM8X	8x oversampling using arithmetic baud rate generation
0x3	FRAC8X	8x oversampling using fractional baud rate generation
0x4	ARITHM3X	3x oversampling using arithmetic baud rate generation

Bit 10 – RXINV Receive Data Invert

Value	Name	Description
0x0	DISABLE	RxD is not inverted
0x1	INV	RxD is inverted

Bit 9 – TXINV Transmit Data Invert

Value	Name	Description
0x0	DISABLE	TxD is not inverted
0x1	INV	TxD is inverted

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

Value	Description
0	STATUS.BUFOVF is asserted when it occurs in the data stream.
1	STATUS.BUFOVF is asserted immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run during Standby

This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

RUNSTDBY	External Clock	Internal Clock
0x0	External clock is disconnected when ongoing transfer is finished. All reception is dropped.	Generic clock is disabled when ongoing transfer is finished. The device will not wake up on either Receive Start or Transfer Complete interrupt unless the appropriate ONDEMAND bits are set in the clocking chain.

RUNSTDBY (continued)		
RUNSTDBY	External Clock	Internal Clock
0x1	Wake on Receive Start or Receive Complete interrupt.	Generic clock is enabled in all sleep modes. Any interrupt can wake up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits select the USART serial communication interface of the SERCOM.
These bits are not synchronized.

Value	Name	Description
0x0	EXTCLK	USART with external clock
0x1	INTCLK	Usart with internal clock

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in a bus error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by the hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.9.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: RW

Write to this register only when SYNCBUSY.CTRLB = 0, otherwise a bus error will result.

Table 34-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							LINCMD[1:0]	
Access							R/W	R/W
Reset							0	0
Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]						RXEN	TXEN
Access	RW	RW					RW	RW
Reset	0	0					0	0
Bit	15	14	13	12	11	10	9	8
			PMODE			ENC	SFDE	COLDEN
Access			RW			RW	RW	RW
Reset			0			0	0	0
Bit	7	6	5	4	3	2	1	0
		SBMODE				CHSIZE[2:0]		
Access		RW				RW	RW	RW
Reset		0				0	0	0

Bits 25:24 – LINCMD[1:0] LIN Command

These bits define the LIN header transmission control. This field is only valid in LIN host mode (CTRLA.FORM= LIN Host).

These are strobe bits and will always read back as zero.

These bits are not enable-protected.

Value	Name	Description
0x0	NONE	Normal USART transmission
0x1	SOFTWARE_CONTROL_TRANSMIT_CMD	Break field is transmitted when DATA is written
0x2	AUTO_TRANSMIT_CMD	Break, synch and identifier are automatically transmitted when DATA is written with the identifier

Bits 23:22 – FIFOCLR[1:0] FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.CTRLB = 0.

These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO

FIFOCLR (continued)		
FIFOCLR[1:0]	Name	Description
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Value	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX and RF FIFOs

Bit 17 – RXEN Receiver Enable

Writing '0' to this bit will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR and BUFOVF bits in the STATUS register.

Writing '1' to CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.

Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or will be enabled when the USART is enabled.

Bit 16 – TXEN Transmitter Enable

Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.

Writing '1' to CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.

Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the transmitter is enabled, and CTRLB.TXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The transmitter is disabled or being enabled.
1	The transmitter is enabled or will be enabled when the USART is enabled.

Bit 13 – PMODE Parity Mode

This bit selects the type of parity used when parity is enabled (CTRLA.FORM is '1'). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.

This bit is not synchronized.

Value	Name	Description
0x0	EVEN	Even parity
0x1	ODD	Odd parity

Bit 10 – ENC Encoding Format

This bit selects the data encoding format.

This bit is not synchronized.

Value	Name	Description
0x0	DISABLE	Data is not encoded
0x1	IRDA	Data is IrDA encoded

Bit 9 – SFDE Start of Frame Detection Enable

This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.

This bit is not synchronized.

SFDE	INTENSET.RXS	INTENSET.RXC	Description
0	X	X	Start-of-frame detection disabled.
1	0	0	Reserved
1	0	1	Start-of-frame detection enabled. RXC wakes up the device from all sleep modes.
1	1	0	Start-of-frame detection enabled. RXS wakes up the device from all sleep modes.
1	1	1	Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes.

Bit 8 – COLDEN Collision Detection Enable

This bit enables collision detection.

This bit is not synchronized.

Value	Description
0	Collision detection is not enabled.
1	Collision detection is enabled.

Bit 6 – SBMODE Stop Bit Mode

This bit selects the number of stop bits transmitted.

This bit is not synchronized.

Value	Name	Description
0x0	ONE	One stop bit
0x1	TWO	Two stop bits

Bits 2:0 – CHSIZE[2:0] Character Size

These bits select the number of bits in a character.

These bits are not synchronized.

CHSIZE[2:0]	Description
0x0	8 bits
0x1	9 bits
0x2-0x4	Reserved
0x5	5 bits
0x6	6 bits
0x7	7 bits

Value	Name	Description
0x0	8BITS	8-bits character
0x1	9BITS	9-bits character
0x5	5BITS	5-bits character
0x6	6BITS	6-bits character
0x7	7BITS	7-bits character

34.9.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: RW

Table 34-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN		DATA32B[1:0]	
Access	RW	RW	RW	RW	RW		R/W	R/W
Reset	0	0	0	0	0		0	0
Bit	23	22	21	20	19	18	17	16
		MAXITER[2:0]					DSNACK	INACK
Access		RW	RW	RW			RW	RW
Reset		0	0	0			0	0
Bit	15	14	13	12	11	10	9	8
					HDRDLY[1:0]		BRKLEN[1:0]	
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
						GTIME[2:0]		
Access						RW	RW	RW
Reset						0	0	0

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold

These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

TXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated as long as the FIFO is not full
0x1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free
0x2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold

These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

RXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.

RXTRHOLD (continued)		
RXTRHOLD	Name	Description
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated when DATA is present in the FIFO
0x1	HALF	Interrupt and DMA triggers are generated when FIFO is half-full
0x2	FULL	Interrupt and DMA triggers are generated when FIFO is full

Bit 27 – FIFOEN FIFO Enable

This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

Bits 25:24 – DATA32B[1:0] Data 32 Bit

These bits configure 32-bit Extension for read and write transactions to the DATA register. When disabled, access is according to CTRLB.CHSIZE.

Value	Description
0x0	DATA reads (for received data) and writes (for transmit data) according to CTRLB.CHSIZE.
0x1	DATA reads according to CTRLB.CHSIZE. DATA writes using 32-bit Extension.
0x2	DATA reads using 32-bit Extension. DATA writes according to CTRLB.CHSIZE.
0x3	DATA reads and writes using 32-bit Extension.

Bits 22:20 – MAXITER[2:0] Maximum Iterations

Bit 17 – DSNACK Disable Successive NACK

Bit 16 – INACK Inhibit Not Acknowledge

Bits 11:10 – HDRDLY[1:0] LIN Master Header Delay

These bits define the delay between break and sync transmission in addition to the delay between the sync and identifier (ID) fields when in LIN host mode (CTRLA.FORM=0x2).

This field is only valid when using the LIN header command (CTRLB.LINCMD=0x2).

Note: A correct setting for this bit field must be selected in order to follow the LIN specification.

Value	Name	Description
0x0	1BIT	Delay between break and sync transmission is 1-bit time
0x1	4BITS	Delay between break and sync transmission is 4-bit time
0x2	8BITS	Delay between break and sync transmission is 8-bit time
0x3	14BITS	Delay between break and sync transmission is 14-bit time

Bits 9:8 – BRKLEN[1:0] LIN Master Break Length

These bits define the length of the break field transmitted when in LIN host mode (CTRLA.FORM=0x2).

Note: A correct setting for this bit field must be selected in order to follow the LIN specification.

Value	Name	Description
0x0	13BITS	Break field transmission is 13 bit times
0x1	17BITS	Break field transmission is 17 bit times
0x2	21BITS	Break field transmission is 21 bit times

Value	Name	Description
0x3	26BITS	Break field transmission is 26 bit times

Bits 2:0 – GTIME[2:0] Guard Time

These bits define the guard time when using RS485 mode (CTRLA.FORM=0x0 or CTRLA.FORM=0x1, and CTRLA.TXPO=0x3).

For RS485 mode, the guard time is programmable from 0-7 bit times and defines the time that the transmit enable pin (TE) remains high after the last stop bit is transmitted and there is no remaining data to be transmitted.

34.9.4 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x0000
Property: RW

Table 34-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	BAUD[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – BAUD[15:0] Baud Rate Value

Arithmetic Baud Rate Generation (CTRLA.SAMPR[0]=0):

These bits control the clock generation, as described in the SERCOM Baud Rate section.

If Fractional Baud Rate Generation (CTRLA.SAMPR[0]=1 or =3) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**

These bits control the clock generation, as described in the section.

- **Bits 12:0 - BAUD[12:0]: Baud Value**

These bits control the clock generation, as described in the section.

34.9.5 Receive Pulse Length

Name: RXPL
Offset: 0x0E
Reset: 0x00
Property: RW

Table 34-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	RXPL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXPL[7:0] Receive Pulse Length

When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period SE_{per} :

$$PULSE \geq (RXPL + 1) \cdot SE_{per}$$

34.9.6 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Combined Error Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Bit 5 – RXBRK Break Received Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

Bit 4 – CTSIC Clear To Send Input Change Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

Bit 3 – RXS Receive Start Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

Bit 2 – RXC Receive Complete Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Bit 1 – TXC Transmit Complete Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Bit 0 – DRE Data Register Empty Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

34.9.7 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Combined Error Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Bit 5 – RXBRK Break Received Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

Bit 4 – CTSIC Clear To Send Input Change Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

Bit 3 – RXS Receive Start Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

Bit 2 – RXC Receive Complete Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Bit 1 – TXC Transmit Complete Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Bit 0 – DRE Data Register Empty Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

34.9.8 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 34-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR		RXBRK	CTSIC	RXS	RXC	TXC	DRE
Access	RW		RW	RW	W	R	RW	R
Reset	0		0	0	0	0	0	0

Bit 7 – ERROR Combined Error Interrupt

This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 5 – RXBRK Break Received Interrupt

This flag is cleared by writing '1' to it.

This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 4 – CTSIC Clear To Send Input Change Interrupt

This flag is cleared by writing a '1' to it.

This flag is set when a change is detected on the CTS pin.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – RXS Receive Start Interrupt

This flag is cleared by writing '1' to it.

This flag is set when a start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Start interrupt flag.

Bit 2 – RXC Receive Complete Interrupt

This flag is cleared by reading the Data register (DATA) or by disabling the receiver.

This flag is set when there are unread data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete Interrupt

This flag is cleared by writing '1' to it or by writing new data to DATA.

This flag is set when the entire frame in the transmit shift register has been shifted out and there are no new data in DATA.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE Data Register Empty Interrupt

This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready to be written.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

34.9.9 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: RW

Table 34-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ITER	TXE	COLL	ISF	CTS	BUFOVF	FERR	PERR
Access	RW	R/W	RW	RW	R	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 7 – ITER Maximum Number of Repetitions Reached

Bit 6 – TXE Transmitter Empty

When CTRLA.FORM is set to LIN host mode, this bit is set when any ongoing transmission is complete and TxDATA is empty.

When CTRLA.FORM is not set to LIN host mode, this bit will always read back as zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	Transmission not complete, transmitter not empty
1	Transmission Complete, transmitter empty

Bit 5 – COLL Collision Detected

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No collisions detected
1	Collision detected

Bit 4 – ISF Inconsistent Sync Field

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when the frame format is set to auto-baud (CTRLA.FORM = 0x4 or 0x5) and a sync field not equal to 0x55 is received.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No inconsistent sync fields

Value	Description
1	Inconsistent sync field received

Bit 3 – CTS Clear To Send

This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).

Value	Description
0	CTS low
1	CTS high

Bit 2 – BUFOVF Buffer Overflow

Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No buffer overflow
1	Buffer Overflow

Bit 1 – FERR Frame Error

Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Frame Errors
1	Frame Error received

Bit 0 – PERR Parity Error

Reading this bit before reading the Data register will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5, or 0x7) and a parity error is detected.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Parity Errors
1	Parity Error detected

34.9.10 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: R

Table 34-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				LENGTH	RXERRCNT	CTRLB	ENABLE	SWRST
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy

Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH will be set until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, a bus error is generated.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 3 – RXERRCNT RXERRCNT Synchronization Busy

Bit 2 – CTRLB CTRLB Synchronization Busy

Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, a bus error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.9.11 Receive Error Count

Name: RXERRCNT
Offset: 0x20
Reset: 0x00
Property: R

Table 34-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

34.9.12 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: RW

Write to this register only when SYNCBUSY.LENGTH = 0, otherwise a bus error will result.

Table 34-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
							LENEN[1:0]	
Access							RW	RW
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	LEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 9:8 – LENEN[1:0] Data Length Enable

In 32-bit Extension mode, this bit field configures the length counter either for transmit or receive transactions.

Value	Name	Description
0x0	DISABLE	Length counter is disabled
0x1	TX	Length counter is enabled for transmit
0x2	RX	Length counter is enabled for receive

Bits 7:0 – LEN[7:0] Data Length

In 32-bit Extension mode, this bit field configures the data length after which the flags INTFLAG.RXC or INTFLAG.DRE are raised.

Value	Description
0x00	Reserved if LENEN=0x1 or LENEN=0x2
0x01–0xFF	Data Length

34.9.13 Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: RW

Table 34-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data Value

Reading these bits will return the contents of the Receive Data register. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS should be read before reading the DATA value in order to get any corresponding error.

Writing these bits will write the Transmit Data register. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set. Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRL.CDATA32B setting.

34.9.14 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: RW

Table 34-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								RW
Reset								0

Bit 0 – DBGSTOP Debug Mode

This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34.9.15 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: R

Table 34-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				TXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 12:8 – RXSPACE[4:0] Rx FIFO Filled Space

These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] Tx FIFO Empty Space

These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

34.9.16 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: RW

This register provides a copy of internal CPU TX and RX FIFO pointers.

Table 34-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						CPURDPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CPUWRPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 11:8 – CPURDPTR[3:0] CPU FIFO Read Pointer

These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] CPU FIFO Write Pointer

These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

34.10 Register Summary - SPI

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		DORD	CPOL	CPHA	FORM[3:0]			
		23:16			DIPO[1:0]				DOPO[1:0]	
		15:8								IBON
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16	FIFOCLR[1:0]						RXEN	
		15:8	AMODE[1:0]		MSEN				SSDE	
		7:0		PLOADEN				CHSIZE[2:0]		
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
		23:16							FMODE	FRMEN
		15:8					IGNTUR	FSPOL	FSLEN	FSES
		7:0			ICSPACE[5:0]					
0x0C	BAUD	7:0	BAUD[7:0]							
0x0D	Reserved									
...										
0x13										
0x14	INTENCLR	7:0	ERROR				SSL	RXC	TXC	DRE
0x15	Reserved									
0x16	INTENSET	7:0	ERROR				SSL	RXC	TXC	DRE
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR				SSL	RXC	TXC	DRE
0x19	Reserved									
0x1A	STATUS	15:8					LENERR			
		7:0					TUR	BUFOVF		
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0				LENGTH		CTRLB	ENABLE	SWRST
0x20	Reserved									
...										
0x21										
0x22	LENGTH	15:8								LENEN
		7:0	LEN[7:0]							
0x24	ADDR	31:24								
		23:16	ADDRMASK[7:0]							
		15:8								
		7:0	ADDR[7:0]							
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C	Reserved									
...										
0x2F										
0x30	DBGCTRL	7:0								DBGSTOP
0x31	Reserved									
...										
0x33										
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

34.10.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 34-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		DORD	CPOL	CPHA	FORM[3:0]			
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
			DIPO[1:0]				DOPO[1:0]	
Access			RW	RW			RW	RW
Reset			0	0			0	0
Bit	15	14	13	12	11	10	9	8
								IBON
Access								RW
Reset								0
Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 30 – DORD Data Order

This bit selects the data order when a character is shifted out from the Shift register.
This bit is not synchronized.

Value	Name	Description
0x0	MSB	MSB is transmitted first
0x1	LSB	LSB is transmitted first

Bit 29 – CPOL Clock Polarity

In combination with the Clock Phase bit (CPHA), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Value	Name	Description
0x0	LOW	SCK is low when idle
0x1	HIGH	SCK is high when idle

Bit 28 – CPHA Clock Phase

In combination with the Clock Polarity bit (CPOL), this bit determines the SPI Transfer mode.
This bit is not synchronized.

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x0	0	0	Rising, sample	Falling, change

CPHA (continued)

Mode	CPOL	CPHA	Leading Edge	Trailing Edge
0x1	0	1	Rising, change	Falling, sample
0x2	1	0	Falling, sample	Rising, change
0x3	1	1	Falling, change	Rising, sample

Value	Name	Description
0x0	LEADING_EDGE	The data is sampled on a leading SCK edge and changed on a trailing SCK edge
0x1	TRAILING_EDGE	The data is sampled on a trailing SCK edge and changed on a leading SCK edge

Bits 27:24 – FORM[3:0] Frame Format

This bit field selects the various frame formats supported by the SPI in Client mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

FORM[3:0]	Name	Description
0x0	SPI	SPI frame
0x1	-	Reserved
0x2	SPI_ADDR	SPI frame with address
0x3-0xF	-	Reserved

Value	Name	Description
0x0	SPI	SPI frame
0x2	ADDR	SPI frame with address

Bits 21:20 – DIPO[1:0] Data In Pinout

These bits define the Data In (DI) pad configurations.

In host operation, DI is MISO.

In client operation, DI is MOSI.

These bits are not synchronized.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Value	Name	Description
0x0	PAD0	SERCOM PAD0 is used as data input
0x1	PAD1	SERCOM PAD1 is used as data input
0x2	PAD2	SERCOM PAD2 is used as data input
0x3	PAD3	SERCOM PAD3 is used as data input

Bits 17:16 – DOPO[1:0] Data Out Pinout

This bit defines the available pad configurations for Data Out (DO) and the Serial Clock (SCK). In client operation, the Client Select ($\overline{SS}$) line is controlled by DOPO, while in host operation the $\overline{SS}$ line is controlled by the port configuration.

In host operation, DO is MOSI.

In client operation, DO is MISO.

These bits are not synchronized.

DOPO	DO	SCK	Client $\overline{SS}$	Host $\overline{SS}$
0x0	PAD[0]	PAD[1]	PAD[2]	PAD[2] Host $\overline{SS}$ pin when MSSEN = 1 otherwise System configuration
0x1	Reserved			
0x2	PAD[3]	PAD[1]	PAD[2]	PAD[2] Host $\overline{SS}$ pin when MSSEN = 1 otherwise System configuration
0x3	Reserved			

Value	Name	Description
0x0	PAD0	DO on PAD[0], SCK on PAD[1] and SS on PAD[2]
0x2	PAD3	DO on PAD[3], SCK on PAD[1] and SS on PAD[2]

Bit 8 – IBON Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.

This bit is not synchronized.

Value	Description
0	STATUS.BUFOVF is set when it occurs in the data stream.
1	STATUS.BUFOVF is set immediately upon buffer overflow.

Bit 7 – RUNSTDBY Run during Standby

This bit defines the functionality in Standby Sleep mode.

This bit is not synchronized.

RUNSTDBY	Client	Host
0x0	Disabled. All reception is dropped, including the ongoing transaction.	Generic clock is disabled when ongoing transaction is finished. All interrupts can wake-up the device.
0x1	Ongoing transaction continues, wake on Receive Complete interrupt.	Generic clock is enabled while in sleep modes. All interrupts can wake-up the device.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x2 or 0x3 to select the SPI of the SERCOM.

0x2: SPI client operation

0x3: SPI host operation

These bits are not synchronized.

Value	Name	Description
0x2	SLAVE	SPI slave operation
0x3	MASTER	SPI master operation

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled or being enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing Reset will result in a bus error. Reading any register will return the Reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete.

CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by the hardware.

Value	Description
0	There is no Reset operation ongoing.
1	The Reset operation is ongoing.

34.10.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: RW

Write to this register only when SYNCBUSY.CTRLB = 0, otherwise a bus error will result.

Table 34-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]						RXEN	
Access	RW	RW					RW	
Reset	0	0					0	
Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]		MSSEN				SSDE	
Access	RW	RW	RW				RW	
Reset	0	0	0				0	
Bit	7	6	5	4	3	2	1	0
		PLOADEN				CHSIZE[2:0]		
Access		RW				RW	RW	RW
Reset		0				0	0	0

Bits 23:22 – FIFOCLR[1:0] FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.CTRLB = 0.

These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Value	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX and RF FIFOs

Bit 17 – RXEN Receiver Enable

Writing '0' to this bit will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.

Writing '1' to this bit when the SPI is disabled will set it immediately. When the SPI is enabled, RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled RXEN will read back as '1'.

Writing '1' to RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and RXEN will read back as '1'.

This bit is not enable-protected.

Value	Description
0	The receiver is disabled or being enabled.
1	The receiver is enabled or it will be enabled when SPI is enabled.

Bits 15:14 – AMODE[1:0] Address Mode

These bits set the Client Addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in Host mode.

AMODE[1:0]	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2_ADDRS	The client responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The client responds to the range of addresses : ADDRMASK < address < ADDR
0x3	-	Reserved

Value	Name	Description
0x0	MASK	ADDRMASK is used as a mask to the ADDR register
0x1	2ADDRS	The slave responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit

Bit 13 – MSSEN Master Slave Select Enable

This bit enables hardware Client Select ($\overline{SS}$) control.

Value	Name	Description
0x0	DISABLE	Hardware SS control is disabled
0x1	ENABLE	Hardware SS control is enabled

Bit 9 – SSDE Slave Select Low Detect Enable

This bit enables wake-up when the Client Select ($\overline{SS}$) pin transitions from high to low.

Value	Name	Description
0x0	DISABLE	SS low detector is disabled
0x1	ENABLE	SS low detector is enabled

Bit 6 – PLOADEN Data Preload Enable

Setting this bit will enable preloading of the Client Shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it will be transferred immediately to the Shift register.

Bits 2:0 – CHSIZE[2:0] Character Size

CHSIZE[2:0]	Name	Description
0x0	8BIT	8 bits
0x1	9BIT	9 bits
0x2-0x7	-	Reserved

Value	Name	Description
0x0	8BITS	8-bits character

Value	Name	Description
0x1	9BITS	9-bits character

34.10.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: RW

Table 34-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
Access	RW	RW	RW	RW	RW			RW
Reset	0	0	0	0	0			0
Bit	23	22	21	20	19	18	17	16
							FMODE	FRMEN
Access							RW	RW
Reset							0	0
Bit	15	14	13	12	11	10	9	8
					IGNTUR	FSPOL	FSLEN	FSES
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
			ICSPACE[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold

These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

TXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated as long as the FIFO is not full
0x1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free
0x2	EMPTY	Interrupt and DMA triggers are generated as long as the FIFO is empty

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold

These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

RXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.

RXTRHOLD (continued)

RXTRHOLD	Name	Description
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated when DATA is present in the FIFO
0x1	HALF	Interrupt and DMA triggers are generated when FIFO is half-full
0x2	FULL	Interrupt and DMA triggers are generated when FIFO is full

Bit 27 – FIFOEN FIFO Enable

This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

Bit 24 – DATA32B Data 32 Bit

This bit enables 32-bit Extension for read and write transactions to the DATA register. When disabled, access is according to CTRLB.CHSIZE.

Value	Description
0	Transactions from and to DATA register are 8-bit
1	Transactions from and to DATA register are 32-bit

Bit 17 – FMODE Frame mode

This bit defines the Frame Mode.

FMODE	Name	Description
0	HOST	Frame Host
1	CLIENT	Frame Client

Value	Name	Description
0x0	MASTER	Frame Master
0x1	SLAVE	Frame Slave

Bit 16 – FRMEN Frame mode enable

This bit enables the SPI Frame Mode operation.

FRMEN	Name	Description
0	DISABLE	Frame Mode Disabled
1	ENABLE	Frame Mode Enabled

Bit 11 – IGTUR Ignore Transmit Underrun

This bit controls when the underrun conditions in framed mode must be ignored. This bit is not synchronized.

Value	Description
0	When a new FSYNC is detected, zero-bytes frames will be sent while underrun condition detection is not cleared.
1	When a new FSYNC is detected, DATA will be transmitted.

Bit 10 – FSPOL Frame Synch Polarity

This bit defines the valid Frame Synch polarity.

FSPOL	Name	Description
0	HIGH	VCC-level valid polarity
1	LOW	Ground-level valid polarity

Value	Name	Description
0x0	HIGH	VCC-level valid polarity
0x1	LOW	GND-level valid polarity

Bit 9 – FSLEN Frame Synch Length

This bit defines the Frame Synch duration.

FSLEN	Name	Description
0	STROBE	One SCK pulse
1	LEVEL	One frame duration valid level

Value	Name	Description
0x0	STROBE	One SCK pulse
0x1	LEVEL	One frame duration

Bit 8 – FSES Frame Synch Edge Select

This bit controls when the frame sync pulse edge must be generated.

This bit is not synchronized.

Value	Description
0	Frame synchronization pulse (idle-to-active edge) precedes the first bit clock.
1	Frame synchronization pulse (idle-to-active edge) coincides with the first bit clock.

Bits 5:0 – ICSpace[5:0] Inter-Character Spacing

When non-zero, CTRLC.ICSPACE selects the minimum number of baud cycles the SCK line will not toggle between characters.

Value	Description
0x00	Inter-Character Spacing is disabled
0x01–0x3F	The minimum Inter-Character Spacing

34.10.4 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00
Property: RW

Table 34-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BAUD[7:0] Baud Rate Value

These bits control the clock generation, as described in the .

34.10.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	RW				RW	RW	RW	RW
Reset	0				0	0	0	0

Bit 7 – ERROR Combined Error Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Bit 3 – SSL Slave Select Low Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Client Select Low Interrupt Enable bit, which disables the Client Select Low interrupt.

Bit 2 – RXC Receive Complete Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

Bit 1 – TXC Transmit Complete Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disable the Transmit Complete interrupt.

Bit 0 – DRE Data Register Empty Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

34.10.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	RW				RW	RW	RW	RW
Reset	0				0	0	0	0

Bit 7 – ERROR Combined Error Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Bit 3 – SSL Slave Select Low Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Client Select Low Interrupt Enable bit, which enables the Client Select Low interrupt.

Bit 2 – RXC Receive Complete Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

Bit 1 – TXC Transmit Complete Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

Bit 0 – DRE Data Register Empty Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

34.10.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 34-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR				SSL	RXC	TXC	DRE
Access	RW				RW	R	RW	R
Reset	0				0	0	0	0

Bit 7 – ERROR Combined Error Interrupt

This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding flags in the STATUS register: TUR error, BUFOVF error, and the LENERR error.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 3 – SSL Slave Select Low Interrupt Flag

This flag is cleared by writing '1' to it.

This bit is set when a high to low transition is detected on the $\overline{SS}$ pin in Client mode and Client Select Low Detect (CTRLB.SSDE) is enabled.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 2 – RXC Receive Complete Interrupt

This flag is cleared by reading the Data (DATA) register or by disabling the receiver.

This flag is set when there are unread data in the receive buffer. If address matching is enabled (CTRLA.FORM = 0x2), the first data received in a transaction will be an address.

Writing '0' to this bit has no effect.

Writing '1' to this bit has no effect.

Bit 1 – TXC Transmit Complete Interrupt

This flag is cleared by writing '1' to it or by writing new data to DATA.

In Host mode, this flag is set when the data have been shifted out and there are no new data in DATA.

In Client mode, this flag is set when the $\overline{SS}$ pin is pulled high. If address matching is enabled (CTRLA.FORM = 0x2), this flag is only set if the transaction was initiated with an address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 0 – DRE Data Register Empty Interrupt

This flag is cleared by writing new data to DATA.

This flag is set when DATA is empty and ready for new data to transmit.

Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

34.10.8 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: RW

Table 34-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
					LENERR			
Access					RW			
Reset					0			

Bit	7	6	5	4	3	2	1	0
					TUR	BUFOVF		
Access					RW	RW		
Reset					0	0		

Bit 11 – LENERR Transaction Length Error

This bit is set in client mode when the length counter is enabled (LENGTH.LENEN=1) and the transfer length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Length Error has occurred.
1	A Length Error has occurred.

Bit 3 – TUR Frame Transmit Underrun

This bit is cleared by writing '1' to the bit or by disabling the receiver. This bit is set when an underflow condition is detected in frame mode. Writing '0' to this bit has no effect. Writing '1' to this bit sends a request to clear it. It will be actually cleared after the DATA buffer is flushed.

Value	Description
0	No Underrun has occurred.
1	A Underrun has occurred.

Bit 2 – BUFOVF Buffer Overflow

Reading this bit before reading DATA will indicate the error status of the next character to be read.

This bit is cleared by writing '1' to the bit or by disabling the receiver.

This bit is set when a Buffer Overflow condition is detected. See also CTRLA.IBON for overflow handling.

When set, the corresponding RxDATA will be zero.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

Value	Description
0	No Buffer Overflow has occurred.
1	A Buffer Overflow has occurred.

34.10.9 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: R

Table 34-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				LENGTH		CTRLB	ENABLE	SWRST
Access				R		R	R	R
Reset				0		0	0	0

Bit 4 – LENGTH LENGTH Synchronization Busy

Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH will be asserted until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, a bus error is generated.

Note: In client mode, the clock is only running during data transfer, so SYNCBUSY.LENGTH will remain asserted until the next data transfer begins.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 2 – CTRLB CTRLB Synchronization Busy

Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB=1 until synchronization is complete, after which the bit will read zero. If CTRLB is written while SYNCBUSY.CTRLB=1, a bus error will be generated.

Value	Description
0	CTRLB synchronization is not busy.
1	CTRLB synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE=1 until synchronization is complete, after which it will read zero.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST=1 until synchronization is complete, after which it will read zero.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.10.10 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: RW

Write to this register only when SYNCBUSY.LENGTH = 0, otherwise a bus error will result.

Table 34-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
								LENEN
Access								RW
Reset								0

Bit	7	6	5	4	3	2	1	0
	LEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 8 – LENEN Data Length Enable

In 32-bit Extension mode, this bit field enables the length counter.

Value	Description
0	Length counter disabled
1	Length counter enabled

Bits 7:0 – LEN[7:0] Data Length

In 32-bit Extension mode, this bit field configures the data length after which the flags INTFLAG.RCX or INTFLAG.DRE are raised.

Value	Description
0x00	Reserved if LENEN=0x1
0x01–0xFF	Data Length

34.10.11 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: RW

Table 34-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	ADDRMASK[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – ADDRMASK[7:0] Address Mask

These bits hold the address mask when the transaction format with address is used (CTRLA.FORM = 0x2, CTRLB.AMODE).

Bits 7:0 – ADDR[7:0] Address Value

These bits hold the address when the transaction format with address is used (CTRLA.FORM = 0x2, CTRLB.AMODE).

34.10.12 Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: RW

Table 34-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data Value

Reading these bits will return the contents of the receive data buffer. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits will write the transmit data buffer. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.

34.10.13 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: RW

Table 34-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								RW
Reset								0

Bit 0 – DBGSTOP Debug Mode

This bit controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34.10.14 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: R

This register allows the user to identify the number of bytes present in each TX and RX FIFO.

Table 34-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				TXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 12:8 – RXSPACE[4:0] Rx FIFO Filled Space

These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] Tx FIFO Empty Space

These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

Note: When preloading is used, data written to shift register is not accessible in TXSPACE.

34.10.15 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: RW

This register provides a copy of internal CPU TX and RX FIFO pointers.

Table 34-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						CPURDPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CPUWRPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 11:8 – CPURDPTR[3:0] CPU FIFO Read Pointer

These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] CPU FIFO Write Pointer

These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

Note: When preloading is used, data written to shift register is not accessible in CPUWRPTR.

34.11 Register Summary - I2CM

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24	LOWTOUTEN		INACTOUT[1:0]		SCLSM		SPEED[1:0]	
		23:16	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]				SMBUSEN	PINOUT
		15:8					SLEWRATE[1:0]		FILTSEL[1:0]	
		7:0	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
		15:8							QCEN	SMEN
		7:0								
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			
		23:16								
		15:8								
		7:0								
0x0C	BAUD	31:24	HSBAUDLOW[7:0]							
		23:16	HSBAUD[7:0]							
		15:8	BAUDLOW[7:0]							
		7:0	BAUD[7:0]							
0x10 ... 0x13	Reserved									
0x14	INTENCLR	7:0	ERROR			RXFF	TXFE		SB	MB
0x15	Reserved									
0x16	INTENSET	7:0	ERROR			RXFF	TXFE		SB	MB
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR			RXFF	TXFE		SB	MB
0x19	Reserved									
0x1A	STATUS	15:8						LENERR	SEXTTOUT	MEXTTOUT
		7:0	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0			LENGTH		SYSOP	ENABLE	SWRST	
0x20 ... 0x23	Reserved									
0x24	ADDR	31:24								
		23:16	LEN[7:0]							
		15:8	TENBITEN	HS	LENEN			ADDR[10:8]		
		7:0	ADDR[7:0]							
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C ... 0x2F	Reserved									
0x30	DBGCTRL	7:0								DBGSTOP
0x31 ... 0x33	Reserved									
0x34	FIFOSPACE	15:8			RXSPACE[4:0]					
		7:0			TXSPACE[4:0]					
0x36	FIFOPTR	15:8				CPURDPTR[3:0]				
		7:0				CPUWRPTR[3:0]				

34.11.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 34-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		LOWTOUTEN	INACTOUT[1:0]		SCLSM		SPEED[1:0]	
Access		RW	RW	RW	RW		RW	RW
Reset		0	0	0	0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN	MEXTTOEN	SDAHOLD[1:0]				SMBUSEN	PINOUT
Access	RW	RW	RW	RW			RW	RW
Reset	0	0	0	0			0	0

Bit	15	14	13	12	11	10	9	8
					SLEWRATE[1:0]		FILTSEL[1:0]	
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUTEN SCL Low Timeout Enable

Bits 29:28 – INACTOUT[1:0] Inactive Time-Out

If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arises when either an I²C host or client is holding the SCL low.

Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.

Calculated time-out periods are based on a 100 kHz baud rate.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disable
0x1	55US	5-6 SCL cycle time-out
0x2	105US	10-11 SCL cycle time-out
0x3	205US	20-21-6 SCL cycle time-out

Bit 27 – SCLSM SCL Clock Stretch Mode

This bit controls when SCL will be stretched for software interaction.

This bit is not synchronized.

Value	Description
0	SCL stretch according to 2C Host Behavioral Diagram (SCLSM = 0) .
1	SCL stretch only after ACK bit, 2C Host Behavioral Diagram (SCLSM = 1) .

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.
These bits are not synchronized.

Value	Name	Description
0x0	SM	Standard-Mode (SM) and Fast-Mode (FM)
0x1	FMP	Fast-Mode Plus (FM+)
0x2	HS	High-Speed Mode

Bit 23 – SEXTTOEN Slave SCL Low Extend Timeout

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the host will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted. SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bit 22 – MEXTTOEN Master SCL Low Extend Timeout

This bit enables the host SCL low extend time-out. If SCL is cumulatively held low for greater than 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the host will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted. SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time

These bits define the SDA hold time with respect to the negative edge of SCL.
These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disable
0x1	75NS	50ns - 100ns hold time
0x2	450NS	300ns - 600ns hold time
0x3	600NS	400ns - 800ns hold time

Bit 17 – SMBUSEN SMBUS Input Buffer Enable

This bit enables SMBus-compatible I/O logic level.
This bit is not synchronized.

Value	Description
0	SMBus input buffer is disabled.
1	SMBus input buffer is enabled.

Bit 16 – PINOUT Pin Usage

This bit set the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled.
1	4-wire operation enabled.

Bits 11:10 – SLEWRATE[1:0] Slew Rate Selection

This bit enables the I/O pins slew rate control.

This bit is not synchronized.

Value	Name	Description
0x0	SMM	Standard mode
0x1	FM	Fast mode
0x2	FMP	Fast mode plus
0x3	HS	High-speed mode

Bits 9:8 – FILTSEL[1:0] Input Filter Selection

These bits define filter length applied to the input signals.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disable
0x2	50NS	Minimum 50ns filter
0x3	10NS	Minimum 10ns filter

Bit 7 – RUNSTDBY Run in Standby

This bit defines the functionality in standby sleep mode.

This bit is not synchronized.

Value	Description
0	GCLK_SERCOMx_CORE is disabled and the I ² C host will not operate in standby sleep mode.
1	GCLK_SERCOMx_CORE is enabled in all sleep modes.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x5 to select the I²C host serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Name	Description
5	I2C_MASTER	I2C host mode operation

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in a bus error. Reading any register will return the reset value of the register.

Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete.

CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by the hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.11.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 34-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
Access	RW	RW				RW	W	W
Reset	0	0				0	0	0
Bit	15	14	13	12	11	10	9	8
							QCEN	SMEN
Access							RW	RW
Reset							0	0
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:22 – FIFOCLR[1:0] FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Value	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX and RF FIFOs

Bit 18 – ACKACT Acknowledge Action

This bit defines the I²C host's acknowledge behavior after a data byte is received from the I²C client. The acknowledge action is executed when a command is written to CTRLB.CMD, or if Smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.

This bit is not enable-protected.
This bit is not write-synchronized.

Value	Description
0	Send ACK.
1	Send NACK.

Bits 17:16 – CMD[1:0] Command

Writing these bits triggers a host operation as described below. The CMD bits are strobe bits, and always read as zero. The acknowledge action is only valid in Host Read mode. In Host Write mode, a command will only result in a repeated Start or Stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and then the acknowledge action will be updated before the command is triggered.

Commands can only be issued when either the Client on Bus Interrupt flag (INTFLAG.SB) or Host on Bus Interrupt flag (INTFLAG.MB) is '1'.

If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.

Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).

Table 34-54. Command Description

CMD[1:0]	Direction	Action
0x0	X	(No action)
0x1	X	Execute acknowledge action succeeded by repeated Start
0x2	0 (Write)	No operation
	1 (Read)	Execute acknowledge action succeeded by a byte read operation
0x3	X	Execute acknowledge action succeeded by issuing a Stop condition

These bits are not enable-protected.

Bit 9 – QCEN Quick Command Enable

This bit is not write-synchronized.

Value	Description
0	Quick Command is disabled.
1	Quick Command is enabled.

Bit 8 – SMEN Smart Mode Enable

When Smart mode is enabled, acknowledge action is sent when DATA.DATA is read.

This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.11.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: RW

Table 34-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
Access	RW	RW	RW	RW	RW			RW
Reset	0	0	0	0	0			0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold

These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

TXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated as long as the FIFO is not full
0x1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free
0x2	EMPTY	Interrupt and DMA triggers are generated as long as the FIFO is not full

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold

These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

RXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.

RXTRHOLD (continued)

RXTRHOLD	Name	Description
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated when DATA is present in the FIFO
0x1	HALF	Interrupt and DMA triggers are generated when FIFO is half-full
0x2	FULL	Interrupt and DMA triggers are generated when FIFO is full

Bit 27 – FIFOEN FIFO Enable

This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

Bit 24 – DATA32B Data 32 Bit

This bit enables 32-bit data writes and reads to/from the DATA register.

Value	Description
0	Data transactions to/from DATA are 8-bit in size
1	Data transactions to/from DATA are 32-bit in size

34.11.4 Baud Rate

Name: BAUD
Offset: 0x0C
Reset: 0x00000000
Property: RW

Table 34-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	HSBAUDLOW[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	HSBAUD[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BAUDLOW[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BAUD[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – HSBAUDLOW[7:0] High Speed Baud Rate Value Low

HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to $HSBAUDLOW = f_{GCLK} \cdot T_{LOW} - 1$
HSBAUDLOW equal to zero: The HSBAUD register is used to time T_{LOW} , T_{HIGH} , $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. T_{BUF} is timed by the BAUD register.

Bits 23:16 – HSBAUD[7:0] High Speed Baud Rate Value

This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, T_{LOW} , T_{HIGH} , $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. T_{BUF} is timed by the BAUD register.
 $HSBAUD = f_{GCLK} \cdot T_{HIGH} - 1$

Bits 15:8 – BAUDLOW[7:0] Baud Rate Value Low

If this bit field is non-zero, the SCL low time will be described by the value written. For more information on how to calculate the frequency, see .

Bits 7:0 – BAUD[7:0] Baud Rate Value

This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL. For more information on how to calculate the frequency, see .

34.11.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE		SB	MB
Access	RW			RW	RW		RW	RW
Reset	0			0	0		0	0

Bit 7 – ERROR Combined Error Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Bit 4 – RXFF Rx FIFO Full Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the RX FIFO Full bit, which disables the RX FIFO Full interrupt.

Bit 3 – TXFE Tx FIFO Empty Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the TX FIFO Empty bit, which disables the TX FIFO Empty interrupt.

Bit 1 – SB Slave On Bus Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Client on Bus Interrupt Enable bit, which disables the Client on Bus interrupt.

Value	Description
0	The Client on Bus interrupt is disabled.
1	The Client on Bus interrupt is enabled.

Bit 0 – MB Master On Bus Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Host on Bus Interrupt Enable bit, which disables the Host on Bus interrupt.

Value	Description
0	The Host on Bus interrupt is disabled.
1	The Host on Bus interrupt is enabled.

34.11.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE		SB	MB
Access	RW			RW	RW		RW	RW
Reset	0			0	0		0	0

Bit 7 – ERROR Combined Error Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Bit 4 – RXFF Rx FIFO Full Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the RX FIFO Full bit, which enables the RX FIFO Full interrupt.

Bit 3 – TXFE Tx FIFO Empty Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the TX FIFO Empty bit, which enables the TX FIFO Empty interrupt.

Bit 1 – SB Slave On Bus Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Client on Bus Interrupt Enable bit, which enables the Client on Bus interrupt.

Bit 0 – MB Master On Bus Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Host on Bus Interrupt Enable bit, which enables the Host on Bus interrupt.

34.11.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 34-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE		SB	MB
Access	RW			R	R		RW	RW
Reset	0			0	0		0	0

Bit 7 – ERROR Combined Error Interrupt

This flag is cleared by writing '1' to it.

This bit is set when any error is detected. Errors that will set this flag have corresponding status bits in the STATUS register: LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 4 – RXFF Rx FIFO Full Interrupt

This flag is set when RX FIFO Threshold locations are fulfilled.

The flag is cleared when the RX FIFO is empty.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the RX FIFO Full interrupt flag.

Bit 3 – TXFE Tx FIFO Empty Interrupt

This flag is set when TX FIFO Threshold locations are available.

The flag is cleared when the TX FIFO is full.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the TX FIFO Empty interrupt flag.

Bit 1 – SB Slave On Bus Interrupt

The Client on Bus flag (SB) is set when a byte is successfully received in Host Read mode, for example, no arbitration lost or bus error occurred during the operation. When this flag is set, the host forces the SCL line low, stretching the I²C clock period. The SCL line will be released and the SB bit will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

Bit 0 – MB Master On Bus Interrupt

This flag is set when a byte is transmitted in Host Write mode. The flag is set regardless of the occurrence of a bus error or an Arbitration Lost condition. MB is also set when arbitration is lost during sending of NACK in Host Read mode, or when issuing a Start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the host forces the SCL line low, stretching the I²C clock period. The SCL line will be released and MB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.

Writing '0' to this bit has no effect.

34.11.8 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: RW

Table 34-60. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						LENERR	SEXTTOUT	MEXTTOUT
Access						RW	RW	RW
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT	BUSSTATE[1:0]			RXNACK	ARBLOST	BUSERR
Access	R	RW	RW	RW		R	RW	RW
Reset	0	0	0	0		0	0	0

Bit 10 – LENERR Length Error

This bit is set when automatic length is used for a DMA and/or 32-bit transaction and the client sends a NACK before ADDR.LEN bytes have been written by the host.
Writing '1' to this bit location will clear STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	No Error
1	Transaction Length Error has occurred.

Bit 9 – SEXTTOUT Slave SCL Low Extend Timeout

This bit is set if a client SCL low extend time-out occurs.
This bit is automatically cleared when writing to the ADDR register.
Writing '1' to this bit location will clear SEXTTOUT. Normal use of the I²C interface does not require the SEXTTOUT flag to be cleared by this method.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	No Time-Out
1	Client SCL Low Time-Out has occurred.

Bit 8 – MEXTTOUT Master SCL Low Extend Timeout

This bit is set if a host SCL low time-out occurs.
Writing '1' to this bit location will clear STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	No Time-Out
1	Host SCL Low Extend Time-Out has occurred

Bit 7 – CLKHOLD Clock Hold

This bit is set when the host is holding the SCL line low, stretching the I²C clock. Software should consider this bit when INTFLAG.SB or INTFLAG.MB is set.
This bit is cleared when the corresponding Interrupt flag is cleared and the next operation is given.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	No Time-Out
1	SCL Low Time-Out has occurred

Bit 6 – LOWTOUT SCL Low Timeout

This bit is set if an SCL low time-out occurs.
Writing '1' to this bit location will clear this bit. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

Bits 5:4 – BUSSTATE[1:0] Bus State

These bits indicate the current I²C Bus state.
When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.
Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.
These bits are write synchronized.

Value	Name	Description
0x0	UNKNOWN	The Bus state is unknown to the I ² C host and will wait for a Stop condition to be detected or wait to be forced into an Idle state by software
0x1	IDLE	The Bus state is waiting for a transaction to be initialized
0x2	OWNER	The I ² C host is the current owner of the bus
0x3	BUSY	Some other I ² C host owns the bus

Bit 2 – RXNACK Received Not Acknowledge

This bit indicates whether the last address or data packet sent was acknowledged or not.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

Value	Description
0	Client responded with ACK.
1	Client responded with NACK.

Bit 1 – ARBLOST Arbitration Lost

This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit, or while issuing a Start or Repeated Start condition on the bus. The Host on Bus Interrupt flag (INTFLAG.MB) will be set when STATUS.ARBLOST is set.
Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.

Value	Description
0	No Error
1	Arbitration Lost

Bit 0 – BUSERR Bus Error

This bit indicates that an illegal Bus condition has occurred on the bus, regardless of bus ownership. An illegal Bus condition is detected if a protocol violating start, repeated start or stop is detected on the I²C bus lines. A Start condition directly followed by a Stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set BUSERR.

If the I²C host is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.

Writing the ADDR.ADDR register will automatically clear the BUSERR flag.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear it.

This bit is not write-synchronized.

Value	Description
0	No Error
1	Bus Error has occurred

34.11.9 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: R

Table 34-61. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				LENGTH		SYSOP	ENABLE	SWRST
Access				R		R	R	R
Reset				0		0	0	0

Bit 4 – LENGTH Length Synchronization Busy

Bit 2 – SYSOP System Operation Synchronization Busy

Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization.

Writing CTRLB.FIFOCLR when the SERCOM is enabled and Smart Mode is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.11.10 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: RW

Table 34-62. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	LEN[7:0]							
Reset	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	TENBITEN	HS	LENEN			ADDR[10:8]		
Reset	RW	RW	RW			RW	RW	RW
Reset	0	0	0			0	0	0
Bit	7	6	5	4	3	2	1	0
Access	ADDR[7:0]							
Reset	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – LEN[7:0] Length

These bits define the transaction length of a DMA and/or 32-bit transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

Bit 15 – TENBITEN Ten Bit Addressing Enable

This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

Value	Description
0	10-bit addressing disabled.
1	10-bit addressing enabled.

Bit 14 – HS High Speed Mode

This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

Value	Description
0	High-speed transfer disabled.
1	High-speed transfer enabled.

Bit 13 – LENEN Length Enable

Value	Description
0	Automatic transfer length disabled.
1	Automatic transfer length enabled.

Bits 10:0 – ADDR[10:0] Address Value

When ADDR is written, the consecutive operation will depend on the bus state:

- UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.
- BUSY: The I²C host will await further operation until the bus becomes IDLE.
- IDLE: The I²C host will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.
- OWNER: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set. STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.

The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the host logic to perform any bus protocol related operations.

Note: The I²C host control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.

34.11.11 Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: RW

Table 34-63. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data Value

The host data register I/O location (DATA) provides access to the host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the host (STATUS.CLKHOLD is set). An exception to this is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers I²C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

Writing or reading DATA.DATA when not in smart mode does not require synchronization.

34.11.12 Debug Control

Name: DBGCTRL
Offset: 0x30
Reset: 0x00
Property: RW

Table 34-64. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
								DBGSTOP
Access								RW
Reset								0

Bit 0 – DBGSTOP Debug Mode

This bit controls functionality when the CPU is halted by an external debugger.

Value	Description
0	The baud-rate generator continues normal operation when the CPU is halted by an external debugger.
1	The baud-rate generator is halted when the CPU is halted by an external debugger.

34.11.13 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: R

Table 34-65. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				TXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 12:8 – RXSPACE[4:0] Rx FIFO Filled Space

These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] Tx FIFO Empty Space

These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

34.11.14 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: RW

Table 34-66. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						CPURDPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CPUWRPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 11:8 – CPURDPTR[3:0] CPU FIFO Read Pointer

These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] CPU FIFO Write Pointer

These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

34.12 Register Summary - I2CS

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24		LOWTOUTEN			SCLSM		SPEED[1:0]	
		23:16	SEXTTOEN		SDAHOLD[1:0]				SMBUSEN	PINOUT
		15:8					SLEWRATE[1:0]		FILTSEL[1:0]	
		7:0	RUNSTDBY			MODE[2:0]		ENABLE	SWRST	
0x04	CTRLB	31:24								
		23:16	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
		15:8	AMODE[1:0]					AACKEN	GCMD	SMEN
		7:0								
0x08	CTRLC	31:24	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
		23:16								
		15:8								
		7:0				SDASETUP[3:0]				
0x0C	Reserved									
...										
0x13										
0x14		INTENCLR	7:0	ERROR			RXFF	TXFE	DRDY	AMATCH
0x15	Reserved									
0x16	INTENSET	7:0	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
0x17	Reserved									
0x18	INTFLAG	7:0	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
0x19	Reserved									
0x1A	STATUS	15:8					LENERR	HS	SEXTTOUT	
		7:0	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
0x1C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0				LENGTH		SYSOP	ENABLE	SWRST
0x20	Reserved									
...										
0x21										
0x22	LENGTH	15:8								LENEN
		7:0	LEN[7:0]							
0x24	ADDR	31:24						ADDRMASK[9:7]		
		23:16	ADDRMASK[6:0]							
		15:8	TENBITEN					ADDR[9:7]		
		7:0	ADDR[6:0]							
0x28	DATA	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
0x2C	Reserved									
...										
0x33										
0x34	FIFOSPACE	15:8				RXSPACE[4:0]				
		7:0				TXSPACE[4:0]				
0x36	FIFOPTR	15:8					CPURDPTR[3:0]			
		7:0					CPUWRPTR[3:0]			

34.12.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 34-67. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		LOWTOUTEN			SCLSM		SPEED[1:0]	
Access		RW			RW		RW	RW
Reset		0			0		0	0

Bit	23	22	21	20	19	18	17	16
	SEXTTOEN		SDAHOLD[1:0]				SMBUSEN	PINOUT
Access	RW		RW	RW			RW	RW
Reset	0		0	0			0	0

Bit	15	14	13	12	11	10	9	8
					SLEWRATE[1:0]		FILTSEL[1:0]	
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RUNSTDBY			MODE[2:0]			ENABLE	SWRST
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 30 – LOWTOUTEN SCL Low Timeout Enable

Bit 27 – SCLSM SCL Clock Stretch Mode

This bit controls when SCL will be stretched for software interaction.
This bit is not synchronized.

Value	Description
0	SCL stretch according to 2C Behavioral Diagram (SCLSM = 0)
1	SCL stretch only after ACK bit according to 2C Client Behavioral Diagram SCLSM = 1

Bits 25:24 – SPEED[1:0] Transfer Speed

These bits define bus speed.
These bits are not synchronized.

Value	Name	Description
0x0	SM	Standard-Mode (SM) and Fast-Mode (FM)
0x1	FMP	Fast-Mode Plus (FM+)
0x2	HS	High-Speed Mode

Bit 23 – SEXTTOEN Slave SCL Low Extend Timeout

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the client will release its clock hold if enabled and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.

This bit is not synchronized.

Value	Description
0	Time-out disabled
1	Time-out enabled

Bits 21:20 – SDAHOLD[1:0] SDA Hold Time

These bits define the SDA hold time with respect to the negative edge of SCL.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disable
0x1	75NS	50ns - 100ns hold time
0x2	450NS	300ns - 600ns hold time
0x3	600NS	400ns - 800ns hold time

Bit 17 – SMBUSEN SMBUS Input Buffer Enable

This bit enables SMBus-compatible I/O logic level.

This bit is not synchronized.

Value	Description
0	SMBus input buffer is disabled.
1	SMBus input buffer is enabled.

Bit 16 – PINOUT Pin Usage

This bit sets the pin usage to either two- or four-wire operation:

This bit is not synchronized.

Value	Description
0	4-wire operation disabled
1	4-wire operation enabled

Bits 11:10 – SLEWRATE[1:0] Slew Rate Selection

This bit enables the I/O pins slew rate control.

These bits are not synchronized.

Note: If an I²C function is enabled on a pin, the corresponding PINCFGn.SLEWLIM for that pin MUST = 0x00.

Value	Name	Description
0x0	SM	Standard mode
0x1	FM	Fast mode
0x2	FMP	Fast mode plus
0x3	HS	High-speed mode

Bits 9:8 – FILTSEL[1:0] Input Filter Selection

These bits define filter length applied to the input signals.

These bits are not synchronized.

Value	Name	Description
0x0	DIS	Disabled
0x1	50F	Minimum 50ns filter (SCL fast mode)
0x2	50E	Minimum 50ns filter (SDA even mode)

Value	Name	Description
0x3	10	Minimum 10ns filter

Bit 7 – RUNSTDBY Run during Standby

This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

Value	Description
0	Disabled – All reception is dropped.
1	Wake on address match, if enabled.

Bits 4:2 – MODE[2:0] Operating Mode

These bits must be written to 0x04 to select the I²C client serial communication interface of the SERCOM.

These bits are not synchronized.

Value	Name	Description
0x4	I2C_SLAVE	I2C client mode operation

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

Value	Description
0	The peripheral is disabled or being disabled.
1	The peripheral is enabled.

Bit 0 – SWRST Software Reset

Writing '0' to this bit has no effect.

Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in a bus error. Reading any register will return the reset value of the register.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete.

CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

This bit is not enable-protected.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by the hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

34.12.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 34-68. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	FIFOCLR[1:0]					ACKACT	CMD[1:0]	
Access	RW	RW				RW	W	W
Reset	0	0				0	0	0
Bit	15	14	13	12	11	10	9	8
	AMODE[1:0]					AACKEN	GCMD	SMEN
Access	RW	RW				RW	RW	RW
Reset	0	0				0	0	0
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:22 – FIFOCLR[1:0] FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

FIFOCLR[1:0]	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX/RX FIFO

Value	Name	Description
0x0	NONE	No action
0x1	TXFIFO	Clear TX FIFO
0x2	RXFIFO	Clear RX FIFO
0x3	BOTH	Clear both TX and RF FIFOs

Bit 18 – ACKACT Acknowledge Action

This bit defines the client's acknowledge behavior after an address or data byte is received from the host. The acknowledge action is executed when a command is written to the CMD bits. If smart

mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.

ACKACT shall not be updated more than once between each peripheral interrupts request. This bit is not enable-protected. This bit is not write-synchronized.

Note:

CTRLB.ACKACT shall not be updated more than once between each peripheral interrupt request.

Value	Description
0	Send ACK
1	Send NACK

Bits 17:16 – CMD[1:0] Command

This bit field triggers the client operation as shown below. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the client interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.

All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.

This bit is not enable-protected. This bit is not write-synchronized.

Table 34-69. Command Description

CMD[1:0]	STATUS.DIR Value	Action
0x0	X	(No action)
0x1	X	(Reserved)
0x2	Used to complete a transaction in response to a data interrupt (DRDY)	
	0 (Host write)	Execute acknowledge action succeeded by waiting for any start (S/Sr) condition
	1 (Host read)	Wait for any start (S/Sr) condition
0x3	Used in response to an address interrupt (AMATCH)	
	0 (Host write)	Execute acknowledge action succeeded by reception of next byte
	1 (Host read)	Execute acknowledge action succeeded by client data interrupt
	Used in response to a data interrupt (DRDY)	
	0 (Host write)	Execute acknowledge action succeeded by reception of next byte
	1 (Host read)	Execute a byte read operation followed by ACK/NACK reception

Bits 15:14 – AMODE[1:0] Address Mode

These bits set the addressing mode.

These bits are not write-synchronized.

Value	Name	Description
0x0	MASK	The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK
0x1	2ADDRES	The slave responds to the two unique addresses in ADDR and ADDRMASK
0x2	RANGE	The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit

Bit 10 – AACKEN Automatic Address Acknowledge

This bit enables the address to be automatically acknowledged if there is an address match.

This bit is not write-synchronized.

Value	Description
0	Automatic acknowledge is disabled.
1	Automatic acknowledge is enabled.

Bit 9 – GCMD PMBus Group Command

This bit enables PMBus group command support. When enabled, the Stop Received interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the client has been addressed since the last STOP condition on the bus.

This bit is not write-synchronized.

Value	Description
0	Group command is disabled.
1	Group command is enabled.

Bit 8 – SMEN Smart Mode Enable

When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.
This bit is not write-synchronized.

Value	Description
0	Smart mode is disabled.
1	Smart mode is enabled.

34.12.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: RW

Table 34-70. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TXTRHOLD[1:0]		RXTRHOLD[1:0]		FIFOEN			DATA32B
Access	RW	RW	RW	RW	RW			RW
Reset	0	0	0	0	0			0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
					SDASETUP[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 31:30 – TXTRHOLD[1:0] Transmit FIFO Threshold

These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

TXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated as long as the FIFO is not full.
1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free.
2	EMPTY	Interrupt and DMA triggers are generated when the FIFO is empty.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated as long as the FIFO is not full
0x1	HALF	Interrupt and DMA triggers are generated when half FIFO space is free
0x2	EMPTY	Interrupt and DMA triggers are generated as long as the FIFO is not full

Bits 29:28 – RXTRHOLD[1:0] Receive FIFO Threshold

These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

RXTRHOLD	Name	Description
0	DEFAULT	Interrupt and DMA triggers can be generated when a DATA is present in the FIFO.
1	HALF	Interrupt and DMA triggers can be generated only when the FIFO is half-full.

RXTRHOLD (continued)		
RXTRHOLD	Name	Description
2	FULL	Interrupt and DMA triggers can be generated only when the FIFO is full.
3	-	Reserved

Value	Name	Description
0x0	DEFAULT	Interrupt and DMA triggers are generated when DATA is present in the FIFO
0x1	HALF	Interrupt and DMA triggers are generated when FIFO is half-full
0x2	FULL	Interrupt and DMA triggers are generated when FIFO is full

Bit 27 – FIFOEN FIFO Enable

This bit enables the FIFO operation.

Value	Description
0	FIFO operation is disabled
1	FIFO operation is enabled

Bit 24 – DATA32B Data 32 Bit

This bit enables 32-bit data writes and reads to/from the DATA register.

Value	Description
0	Data transaction to/from DATA are 8-bit in size
1	Data transaction to/from DATA are 32-bit in size

Bits 3:0 – SDASETUP[3:0] SDA Setup Time

These bits select the minimum SDA-to-SCL setup time, measured from the release of SDA to the release of SCL:

$t_{SU:DAT} = (6 + 16 * SDASETUP) \text{ APB CLOCK\_SERCOM periods.}$

34.12.4 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x14
Reset: 0x00
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-71. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – ERROR Combined Error Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

Bit 4 – RXFF Rx FIFO Full Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the RX FIFO Full bit, which disables the RX FIFO Full interrupt.

Bit 3 – TXFE Tx FIFO Empty Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the TX FIFO Empty bit, which disables the TX FIFO Empty interrupt.

Bit 2 – DRDY Data Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready bit, which disables the Data Ready interrupt.

Bit 1 – AMATCH Address Match Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

Bit 0 – PREC Stop Received Interrupt Disable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

34.12.5 Interrupt Enable Set

Name: INTENSET
Offset: 0x16
Reset: 0x00
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

Table 34-72. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – ERROR Combined Error Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

Bit 4 – RXFF Rx FIFO Full Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the RX FIFO Full bit, which enables the RX FIFO Full interrupt.

Bit 3 – TXFE Tx FIFO Empty Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the TX FIFO Empty bit, which enables the TX FIFO Empty interrupt.

Bit 2 – DRDY Data Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

Bit 1 – AMATCH Address Match Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

Bit 0 – PREC Stop Received Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will set the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

34.12.6 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

Table 34-73. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	ERROR			RXFF	TXFE	DRDY	AMATCH	PREC
Access	RW			R	RW	RW	RW	RW
Reset	0			0	0	0	0	0

Bit 7 – ERROR Combined Error Interrupt

This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register: LENERR, SEXTTOUT, LOWTOUT, COLL, and BUSERR.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the flag.

Bit 4 – RXFF Rx FIFO Full Interrupt

This flag is set when RX FIFO Threshold locations are fulfilled.

The flag is cleared when the RX FIFO is empty.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the RX FIFO Full interrupt flag.

Bit 3 – TXFE Tx FIFO Empty Interrupt

This flag is set when TX FIFO Threshold locations are available.

The flag is cleared when the TX FIFO is full.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the TX FIFO Empty interrupt flag.

Bit 2 – DRDY Data Interrupt

This flag is set when a I²C client byte transmission is successfully completed.

The flag is cleared by hardware by either:

- Writing to the DATA register.
- Reading the DATA register with Smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Data Ready Interrupt flag.

Bit 1 – AMATCH Address Match Interrupt

This flag is set when the I²C client address match logic detects that a valid address has been received.

The flag is cleared by hardware when CTRL.CMD is written.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Address Match Interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

Bit 0 – PREC Stop Received Interrupt

This flag is set when a Stop condition is detected for a transaction being processed. A Stop condition detected between a bus host and another client will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).

This flag is cleared by hardware after a command is issued on the next address match.

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Stop Received Interrupt flag.

34.12.7 Status

Name: STATUS
Offset: 0x1A
Reset: 0x0000
Property: RW

Table 34-74. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
					LENERR	HS	SEXTTOUT	
Access					RW	RW	RW	
Reset					0	0	0	

Bit	7	6	5	4	3	2	1	0
	CLKHOLD	LOWTOUT		SR	DIR	RXNACK	COLL	BUSERR
Access	R	RW		R	R	R	RW	RW
Reset	0	0		0	0	0	0	0

Bit 11 – LENERR Transaction Length Error

This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.

This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Bit 10 – HS High Speed

This bit is set if the client detects a START followed by a Host Code transmission.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

Bit 9 – SEXTTOUT Slave SCL Low Extend Timeout

This bit is set if a client SCL low extend time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low extend time-out has occurred.
1	SCL low extend time-out has occurred.

Bit 7 – CLKHOLD Clock Hold

The client Clock Hold bit (STATUS.CLKHOLD) is set when the client is holding the SCL line low, stretching the I²C clock. Software should consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.

This bit is automatically cleared when the corresponding interrupt is also cleared.

Bit 6 – LOWTOUT SCL Low Timeout

This bit is set if an SCL low time-out occurs.

This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 0x3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No SCL low time-out has occurred.
1	SCL low time-out has occurred.

Bit 4 – SR Repeated Start

When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.

This flag is only valid while the INTFLAG.AMATCH flag is one.

Value	Description
0	Start condition on last address match
1	Repeated start condition on last address match

Bit 3 – DIR Read/Write Direction

The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a host.

Value	Description
0	Host write operation is in progress.
1	Host read operation is in progress.

Bit 2 – RXNACK Received Not Acknowledge

This bit indicates whether the last data packet sent was acknowledged or not.

Value	Description
0	Host responded with ACK.
1	Host responded with NACK.

Bit 1 – COLL Transmit Collision

If set, the I²C client was not able to transmit a high data or NACK bit, the I²C client immediately released the SDA and SCL lines and waited for the next packet addressed to it.

This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and should be treated as a bus error.

Note that this status will not trigger any interrupt, and should be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the status.

Value	Description
0	No collision detected on last data byte sent.
1	Collision detected on last data byte sent.

Bit 0 – BUSERR Bus Error

The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start, or stop is detected on the I²C bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set this bit.

This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.

Writing a '1' to this bit will clear the status.

Writing a '0' to this bit has no effect.

Value	Description
0	No bus error detected.
1	Bus error detected.

34.12.8 Synchronization Busy

Name: SYNCBUSY
Offset: 0x1C
Reset: 0x00000000
Property: R

Table 34-75. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				LENGTH		SYSOP	ENABLE	SWRST
Access				R		R	R	R
Reset				0		0	0	0

Bit 4 – LENGTH Length Synchronization Busy

Writing the LENGTH register requires synchronization. When written, this bit will be set until synchronization is complete. If LENGTH is written while SYNCBUSY.LENGTH is asserted, a bus error will be generated.

Note: In client mode, the clock is only running during data transfer, so SYNCBUSY.LENGTH will remain asserted until the next data transfer begins.

Value	Description
0	LENGTH synchronization is not busy.
1	LENGTH synchronization is busy.

Bit 2 – SYSOP System Operation Synchronization Busy

Writing CTRLB.FIFOCLR when the SERCOM is enabled requires synchronization. In this case the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

Value	Description
0	System operation synchronization is not busy.
1	System operation synchronization is busy.

Bit 1 – ENABLE SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

Value	Description
0	Enable synchronization is not busy.
1	Enable synchronization is busy.

Bit 0 – SWRST Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

34.12.9 Length

Name: LENGTH
Offset: 0x22
Reset: 0x0000
Property: RW

Table 34-76. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
								LENEN
Access								RW
Reset								0

Bit	7	6	5	4	3	2	1	0
	LEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 8 – LENEN Data Length Enable

In 32-bit Extension mode (CTRLC.DATA32B=1), this bit field enables the length counter.

Value	Description
0	Length counter is disabled.
1	Length counter is enabled.

Bits 7:0 – LEN[7:0] Data Length

In 32-bit Extension mode (CTRLC.DATA32B=1) with Data Length counting enabled (LENGTH.LENEN), this bit field configures the data length from 0 to 255 Bytes after which the flag INTFLAG.DRDY is raised.

34.12.10 Address

Name: ADDR
Offset: 0x24
Reset: 0x00000000
Property: RW

Table 34-77. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						ADDRMASK[9:7]		
Access						RW	RW	RW
Reset						0	0	0
Bit	23	22	21	20	19	18	17	16
	ADDRMASK[6:0]							
Access	RW	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8
	TENBITEN					ADDR[9:7]		
Access	RW					RW	RW	RW
Reset	0					0	0	0
Bit	7	6	5	4	3	2	1	0
	ADDR[6:0]							GENCEN
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 26:17 – ADDRMASK[9:0] Address Mask

These bits act as a second address match register, an address mask register, or the lower limit of an address range, depending on the CTRLB.AMODE setting.

Bit 15 – TENBITEN Ten Bit Addressing Enable

Value	Description
0	10-bit address recognition disabled.
1	10-bit address recognition enabled.

Bits 10:1 – ADDR[9:0] Address Value

These bits contain the I²C client address used by the client address match logic to determine if a host has addressed the client.

When using 7-bit addressing, the client address is represented by ADDR[6:0].

When using 10-bit addressing (ADDR.TENBITEN=1), the client address is represented by ADDR[9:0].

When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

Bit 0 – GENCEN General Call Address Enable

A general call address is an address consisting of all-zeroes, including the direction bit (host write).

Value	Description
0	General call address recognition disabled.
1	General call address recognition enabled.

34.12.11 Data

Name: DATA
Offset: 0x28
Reset: 0x00000000
Property: RW

Table 34-78. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Data Value

The client data register I/O location (DATA.DATA) provides access to the host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the client (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers I²C bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.

34.12.12 FIFO Space

Name: FIFOSPACE
Offset: 0x34
Reset: 0x0000
Property: R

Table 34-79. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
				RXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
				TXSPACE[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bits 12:8 – RXSPACE[4:0] Rx FIFO Filled Space

These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

Bits 4:0 – TXSPACE[4:0] Tx FIFO Empty Space

These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRL.C.DATA32B setting).

34.12.13 FIFO CPU Pointers

Name: FIFOPTR
Offset: 0x36
Reset: 0x0000
Property: RW

Table 34-80. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
						CPURDPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
						CPUWRPTR[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 11:8 – CPURDPTR[3:0] CPU FIFO Read Pointer

These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

Bits 3:0 – CPUWRPTR[3:0] CPU FIFO Write Pointer

These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.

35. Controller Area Network (CAN)

35.1 Overview

The CAN communication controller provides hardware support for the CAN protocol version 2.0 part A and B and ISO 11898-1:2015 which includes the CAN Flexible Data Rate Frame format (CAN-FD). The CAN module uses system RAM, which it accesses through a Host AHB bus for FIFO and filter storage.

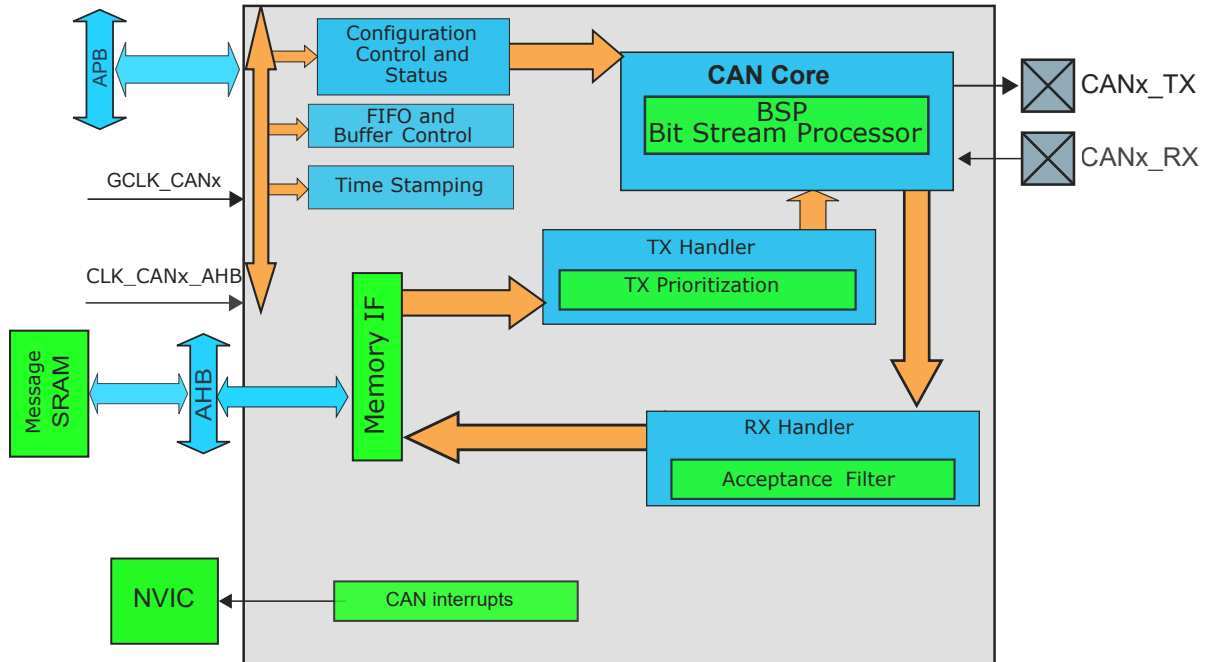
35.2 Features

The following are key features of the CAN module:

- Conforms with CAN protocol version 2.0 part A and B and ISO 11898-1:2015
- Up to two Controller Area Network (CAN)
 - Supporting CAN 2.0 A/B and CAN-FD (ISO 11898-1:2015)
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- Acceptance filtering
 - Up to 128 configurable filter elements
 - Special filter for SAE J1939 IDs
 - Separate signaling on reception of High Priority Messages
- AUTOSAR optimized
- SAE J1939 optimized
- Two configurable Receive FIFOs
- Separate signaling on reception of High-Priority Messages
- Up to 64 dedicated Receive Buffers and up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, Transmit Queue, Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable Loop-Back Test mode
- Maskable module interrupts
- Power-down support; Debug on CAN support
- Transfer rates:
 - 1 Mb/s for CAN 2.0 mode
 - 10 Mb/s for CAN-FD mode
- CiA 603 hardware timestamping support
- External Timestamping Unit (TSU)

35.3 Block Diagram

Figure 35-1. CAN Block Diagram



Note: x= 0,1. This device family contains two CAN modules.

35.4 Signal Description

Table 35-1. Signal Description

PIN NAME	DESCRIPTION	TYPE
CAN0_RX	CAN0_FD Receive	Digital Input
CAN0_TX	CAN0_FD Transmit	Digital Output
CAN1_RX	CAN1_FD Receive	Digital Input
CAN1_TX	CAN1_FD Transmit	Digital Output

35.5 Functional Description

35.5.1 Principle of Operation

The CAN performs communication according to ISO 11898-1:2015 (identical to Bosch CAN protocol specification 2.0 part A,B, ISO CAN FD).

For FIFO and filter storage, the CAN module uses system RAM, which it accesses through a host AHB bus. Each CAN FD instance may use up to 4864 bytes of system memory.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 128 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

35.5.2 Operating Modes

35.5.2.1 Software Initialization

Software initialization is started by setting the CCCR.INIT bit (CCCR <0>), either by software, or by going "bus off." While the CCCR.INIT bit (CCCR <0>) is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output CANx_TX is "recessive" (HIGH). The counters of the Error Management Logic EML are unchanged. Setting the CCCR.INIT bit (CCCR <0>) does not change any configuration register. Resetting the CCCR.INIT bit (CCCR <0>) finishes the software initialization. Afterwards the Bit Stream Processor (BSP) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive "recessive" bits (= Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN configuration registers is only enabled when the CCCR.CCE bit (CCCR <1>) and the CCCR.INIT bit (CCCR <0>) are set (protected write).

The CCCR.CCE bit (CCCR <1>) can only be set and cleared while the CCCR.INIT bit (CCCR <0>) = '1'. The CCCR.CCE bit (CCCR <1>) is automatically cleared when the CCCR.INIT bit (CCCR <0>) is cleared.

The following registers are reset when the CCCR.CCE bit (CCCR <1>) is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value (TOCV.TOC) bits (TOCV <15:0>) is preset to the value configured by the TOCC.TOP bits (TOCC <31:16>) when the CCCR.CCE bit (CCCR <1>) is set.

In addition, the state machines of the Tx Handler and Rx Handler are held in idle state while the CCCR.CCE bit (CCCR <1>) = '1'.

The following registers are only writable while the CCCR.CCE bit (CCCR <1>) = '0'.

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

The CCCR.TEST bit (CCCR <7>) and CCCR.MON bit (CCCR <5>) can only be set by the CPU while the CCCR.INIT bit (CCCR <0>) = '1' and CCCR.CCE bit (CCCR <1>) = '1'. Both bits may be cleared at any time. The CCCR.DAR bit (CCCR <6>) can only be set/cleared while the CCCR.INIT bit (CCCR <0>) = '1' and the CCCR.CCE bit (CCCR <1>) = '1'.

35.5.2.2 Normal Operation

Once the CAN is initialized and CCCR.INIT bit (CCCR <0>) is reset to '0', the CAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO0 or Rx FIFO1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

35.5.2.3 CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD

frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in standard CAN frames with 11-bit identifiers and the first previously reserved bit in extended CAN frames with 29-bit identifiers will now be decoded as FDF bit. A recessive FDF signifies a CAN FD frame, where as a dominant FDF bit signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, (i.e., res and BRS), decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by a dominant res bit and a recessive BRS bit. The coding of a recessive res bit is reserved for future expansion of the protocol. In case the CAN receives a frame with a recessive FDF bit and a recessive res bit, it will signal a Protocol Exception Event by setting bit PSR.PXE bit (PSR <14>). When Protocol Exception Handling is enabled (CCCR.PXHD bit (CCCR<12>) = '0'), this causes the operation state to change from Receiver (PSR.ACT = "0x2") to synchronizing on CAN communication (PSR.ACT bits (PSR <4:3>) = "0x0") at the next sample point. In case Protocol Exception Handling is disabled (CCCR.PXHD bit (CCCR<12>) = '1'), the CAN will treat a recessive res bit as a form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE bit (CCCR<8>). In case CCCR.FDOE bit (CCCR<8>) = '1', transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via FDF bit in the respective Tx Buffer element. With CCCR.FDOE bit (CCCR<8>) = '0', received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if FDF bit of a Tx Buffer element is set. CCCR.FDOE bit (CCCR<8>) and CCCR.BRSE bit (CCCR<9>) can only be changed while CCCR.INIT bit (CCCR <0>) and CCCR.CCE (CCCR <1>) are both set.

With CCCR.FDOE bit (CCCR<8>) = '0', the setting of FDF and BRS bits is ignored and frames are transmitted in Classic CAN format. With CCCR.FDOE bit (CCCR<8>) = '1' and CCCR.BRSE bit (CCCR<9>) = '0', only bit FDF of a Tx Buffer element is evaluated. With CCCR.FDOE bit (CCCR<8>) = '1' and CCCR.BRSE bit (CCCR<9>) = '1', transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with FDF and BRS bits set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN. However, the codes 9 to 15, (which in standard CAN all code a data field of 8 bytes), are coded according to the following table.

Table 35-2. Coding of DLC in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing & Prescaler Register (NBTP). In the following CAN FD data phase, the fast CAN bit timing is used as defined by the Data Bit Timing & Prescaler

Register (DBTP). The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (GCLK_CANx). Example: with a CAN clock frequency of 20MHz and the shortest configurable bit time of $4 t_q$, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

35.5.2.4 Transceiver Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANx_TX the CAN receives the transmitted data from its local CAN transceiver via pin CANx_RX. The received data is delayed by the CAN transceiver's loop delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceivers loop delay.

Description

The CAN's protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point SSP. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting the DBTP.TDC bit (DBTP<23>).

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the CAN's transmit output CANx_TX through the transceiver to the receive input CANx_RX plus the transmitter delay compensation offset as configured by TDCR.TDCO bits (TDCR <15:8>). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of minimum time quantum (mtq). 1 mtq is equal to time period of GCLK_CANx clock.

The PSR.TDCV bits (PSR <22:16>) show the actual transmitter delay compensation value. PSR.TDCV bits (PSR <22:16>) are cleared when CCCR.INIT bit (CCCR <0>) is set and is updated at each transmission of an FD frame while DBTP.TDC bit (DBTP <23>) is set.

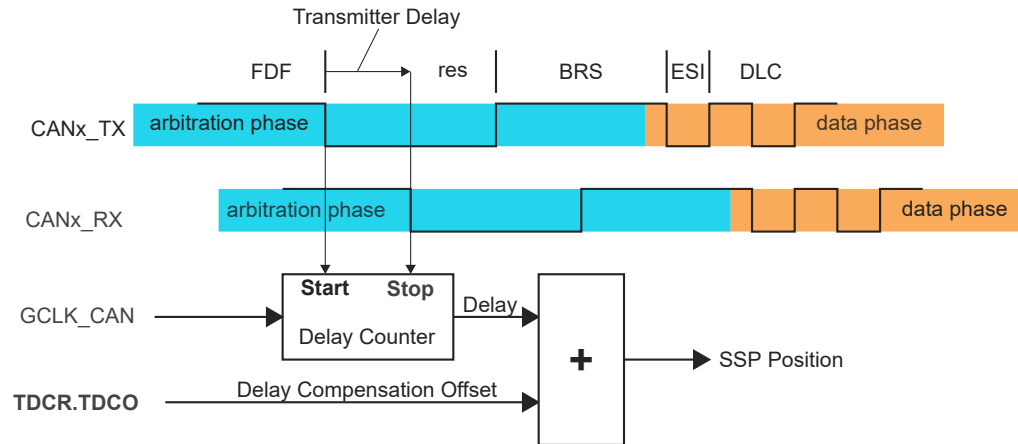
The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN:

- The sum of the measured delay from CANx_TX to CANx_RX and the configured transceiver delay compensation offset FBTP.TDCR bits (TDCR <15:8>) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANx_TX to CANx_RX and the configured transceiver delay compensation offset FBTP.TDCR bits (TDCR <15:8>) has to be less or equal to 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transceiver delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

Transmitter Delay Compensation Measurement

If transmitter delay compensation is enabled by programming the DBTP.TDC bit (DBTP <23>) = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANx_TX of the transmitter. The resolution of this measurement is one mtq.

Figure 35-2. Transceiver delay measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCR.TDCF bit (TDCR <6:0>). This defines a minimum value for the SSP position. Dominant edges of CANx_RX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCR.TDCF bit (TDCR <6:0>) and CANx_RX is low.

35.5.2.5 Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (ECR.REC bit (ECR <14:8>), ECR.TEC bit (ECR <7:0>)) are frozen while Error Logging (ECR.CEL bit (ECR <23:16>)) is still incremented. The CPU can set the CAN into Restricted Operation mode by setting bit CCCR.ASM bit (CCCR <2>). The bit can only be set by the CPU when both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set to '1'. The bit can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU has to reset CCCR.ASM bit (CCCR <2>).

The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

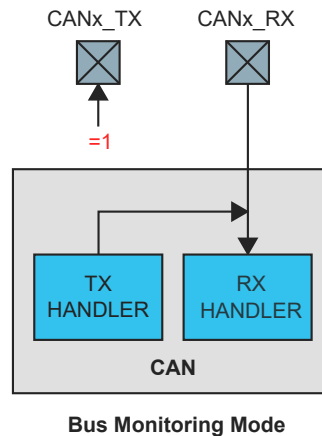
35.5.2.6 Bus Monitoring Mode

The CAN is set in Bus Monitoring Mode by programming CCCR.MON bit (CCCR <5>) to '1'. In Bus Monitoring Mode (see ISO 11898-1, 10.12 Bus monitoring), the CAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN is required to send a dominant bit (ACK bit, overload

flag, active error flag), the bit is rerouted internally so that the CAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode register TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The figure below shows the connection of signals CANx_TX and CANx_RX to the CAN in Bus Monitoring Mode.

Figure 35-3. Pin Control in Bus Monitoring Mode



35.5.2.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO 11898-1, 6.3.3 Recovery Management), the CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via CCCR.DAR bit (CCCR <6>).

Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit (TXBRP.TRPx) is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
 - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
 - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx not set
- Successful transmission in spite of cancellation:
 - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
 - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
 - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx not set
 - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = "10" (transmission in spite of cancellation).

35.5.2.8 Test Modes

To enable write access to register TEST, bit CCCR.TEST bit (CCCR <7>) has to be set to '1'. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANx_TX by programming TEST.TX bits (TEST <6:5>). Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the CAN's bit timing and it can drive constant dominant or recessive values. The actual value at pin CANx_RX can be read from TEST.RX bit (TEST <7>). Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between GCLK_CANx and CLK_CANx_AHB domains, there may be a delay of several CLK_CANx_AHB periods between writing to TEST.TX bits (TEST <6:5>) until the new configuration is visible at output pin CANx_TX. This applies also when reading input pin CANx_RX via TEST.RX bit (TEST <7>).

Note: Test modes should be used for production tests or self test only. The software control for pin CANx_TX interferes with all CAN protocol functions. It is not recommended to use test modes for application.

External Loop Back Mode

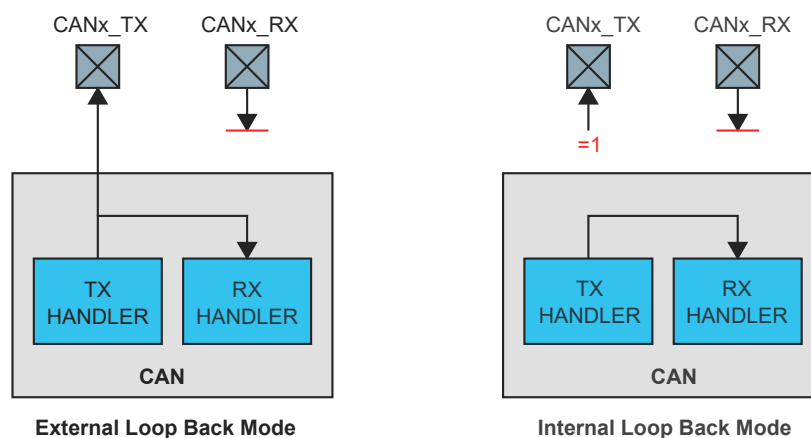
The CAN can be set in External Loop Back Mode by programming TEST.LBCK bit (TEST <4>) to '1'. In Loop Back Mode, the CAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of signals CANx_TX and CANx_RX to the CAN in External Loop Back Mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remot frame) in Loop Back Mode. In this mode the CAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CANx_RX input pin is disregarded by the CAN. The transmitted messages can be monitored at the CANx_TX pin.

Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits TEST.LBCK bit (TEST <4>) and CCCR.MON bit (CCCR <5>) to '1'. This mode can be used for a "Hot Selftest", meaning the CAN can be tested without affecting a running CAN system connected to the pins CANx_TX and CANx_RX. In this mode pin CANx_RX is disconnected from the CAN and pin CANx_TX is held recessive. The following figure shows the connection of CANx_TX and CANx_RX to the CAN in case of Internal Loop Back Mode.

Figure 35-4. Pin Control in Loop Back Modes



35.5.3 Timestamp Generation

35.5.3.1 Default Behavior

For timestamp generation the CAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP bit (TSCC <19:16>) can be configured to clock the counter in multiples of CAN bit times (1...16). The

counter is readable via TSCV.TSC bit (TSCV <15:8>). A write access to register TSCV resets the counter to zero. When the timestamp counter wraps around interrupt flag IR.TSW bit (TSW <16>) is set.

On start of frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

By programming the TSCC.TSS (TSCC <1:0>) bit, a 16-bit external Timestamping Unit (TSU) can be used, instead of the internal 16-bit counter.

35.5.3.2 Behavior When Enabling the TSU

Overview

The TSU implements the requirements specified in CiA 603. This approach is independent of interrupt response times and thus achieves the best possible accuracy. The TSU may operate on its own internal timebase, or use an external time base, e.g. a reference time base from a timer/counter within the device.

The TSU offers the capability to store up to 16 captured timestamps. The resolution depends on the internal or external counter resolution. Up to 32-bit resolution is available.

To enable the CAN-FD hardware timestamping together with the TSU, the CCCR.UTSU (CCCR <10>) bit must be set to 1. If the bit is cleared, the internal 16-bit timestamp counter will be used by default.

A message timestamping can happen at the start-of-frame (SoF), or end-of-frame (EoF).

The TSU is configured via the TSCFG register, which is writable only when the CCCR.CCE (CCCR <1>) = 1.

With TSCFG.TSUE (TSCFG <0>) the TSU is enabled. Default after reset is disabled. The Timebase Prescaler TSCFG.TBPRES (TSCFG <15:8>) is used to adapt the count rate of the TSU's internal counter, if enabled, to the requirements of the application.

Either the internal timebase or an external timebase input can be selected for timestamping via TSCFG.TBCS (TSCFG <1>). In case TSCFG.TBCS (TSCFG <1>) = '0', the internal free running timebase counter is used. In case TSCFG.TBCS (TSCFG <1>) = '1', the timestamp value is taken from the Timebase Input tsu_tbin[31:0].

Reception of Timestamped Messages

To configure the M_CAN for reception of timestamped messages, a Standard/Extended Message ID Filter Element has to be set up by configuration of S0.SSYNC = '1' respectively F1.ESYNC = '1'. In case the filter element matches the received frame and the received frame is valid, a capture will be performed at the EOF. The value of the Actual Timebase (ATB) is stored into the TSU's timestamp register selected by the TSS2.TSP (TSS2 <3:0>). The TSP value from the TSU will be written into the R1.RXTSP of the related RX Buffer or RX FIFO element.

Transmission of Timestamped Messages

In case bits T1.TSCE and T1.EFC of a TX Buffer element are set, a successful transmission of that TX Buffer triggers the storage of the value of the TSU's Actual Timebase ATB to the TSU's timestamp register selected by the Timestamp Pointer TSS2.TSP (TSS2 <3:0>). The Timestamp Pointer value from the TSU is written to the E1.TXTSP of the related TX Event FIFO element.

Capture Point

TSCFG.SCP configures the point in time when a timestamp is captured for Sync messages (0: at EOF, 1: at SOF of the received/transmitted Sync message).

When TSCFG.SCP (TSCFG <2>) = '0', the timestamp is captured directly from the Actual Timebase ATB to the timestamp register TS_n, when the message gets valid at EOF. This configuration is intended for AUTOSAR conformant applications.

When TSCFG.SCP (TSCFG <2>) = '1', the Actual Timebase ATB is captured into a temporary buffer at each SOF.

When the message gets valid at EOF the timestamp is copied from the temporary buffer to the timestamp register TS_n.

Handling of Timestamps

The timestamps are written from TS0 to TS7 in a cyclic manner. Each timestamp register has assigned two status bits:

- TSS1.TSN_n (TSS1 <31:16>) is set whenever a timestamp was stored to the related timestamp register TS_n
- TSS1.TSL_n (TSS1 <15:0>) is set when a new timestamp was stored to the related timestamp register TS_n before the previously store timestamp has been read out

Reading a Timestamp register resets the related TSS1 bits.

Synchronization Process After AUTOSAR

After successful configuration of all participating CAN nodes, the timers of the time slaves can be corrected with a two-step synchronization process. In the first step, the current time T0 is latched in the time master, and the part of T0, which represents the seconds, is sent to the time slaves with the SYNC message. If the EoF (End-Of-Frame) is reached when sending this SYNC message, then timestamps are stored in the TSUs of all nodes, i.e. in the Time Master and all Time Slaves. This is the common point of time reference. In the second step, the time master adds its TSU time stamp to the nanosecond part of T0 and sends it to the time slaves via FollowUp message. From this, the time slaves can derive their own time base.

Saving Time Stamps at SoF (Start-of-Frame)

For non-AUTOSAR applications, the TSU can also be configured to store timestamps at the start of a CAN message (SoF bit).

35.5.4 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the CAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP bit (TSCC <19:16>) as the Timestamp Counter. The Timeout Counter is configured via register TOCC. The actual counter value can be read from TOCV.TOC bits (TOCV <15:0>). The Timeout Counter can only be started while CCCR.INIT bit (CCCR <0>) = '0'. It is stopped when CCCR.INIT bit (CCCR <0>) = '1', e.g. when the CAN enters "bus off" state.

The operation mode is selected by TOCC.TOS bits (TOCC <2:1>). When operating in Continuous Mode, the counter starts when CCCR.INIT bit (CCCR <0>) is reset. A write to TOCV presets the counter to the value configured by TOCC.TOP bits (TOCC <31:16>) and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP bits (TOCC <31:16>). Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, interrupt flag IR.TOO bit (IR <18>) is set. In Continuous Mode, the counter is immediately restarted at TOCC.TOP bits (TOCC <31:16>).

Note: The clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

35.5.5 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

35.5.5.1 Acceptance Filtering

The CAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
 - range filter (from - to)
 - filter for one or two dedicated IDs
 - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM (IR <8>)
- Set High Priority Message interrupt flag IR.HPM (IR <8>) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

Rx Buffer

New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see PSR.LEC bits (PSR <2:0>) respectively PSR.DLEC bit field (PSR<10:8>).

Rx FIFO

Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC bits (PSR <2:0>) respectively PSR.DLEC bit field (PSR<10:8>). In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in *Rx FIFO Overwrite Mode* have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID for standard frames or EF1ID/EF2ID for extended frames.

There are two possibilities when range filtering is used together with extended frames:

- EFT = "00" The Message ID of received frames is AND'ed with the Extended ID AND Mask (XIDAM) before the range filter is applied
- EFT = "11" The Extended ID AND Mask (XIDAM) is not used for range filtering

Filter for specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

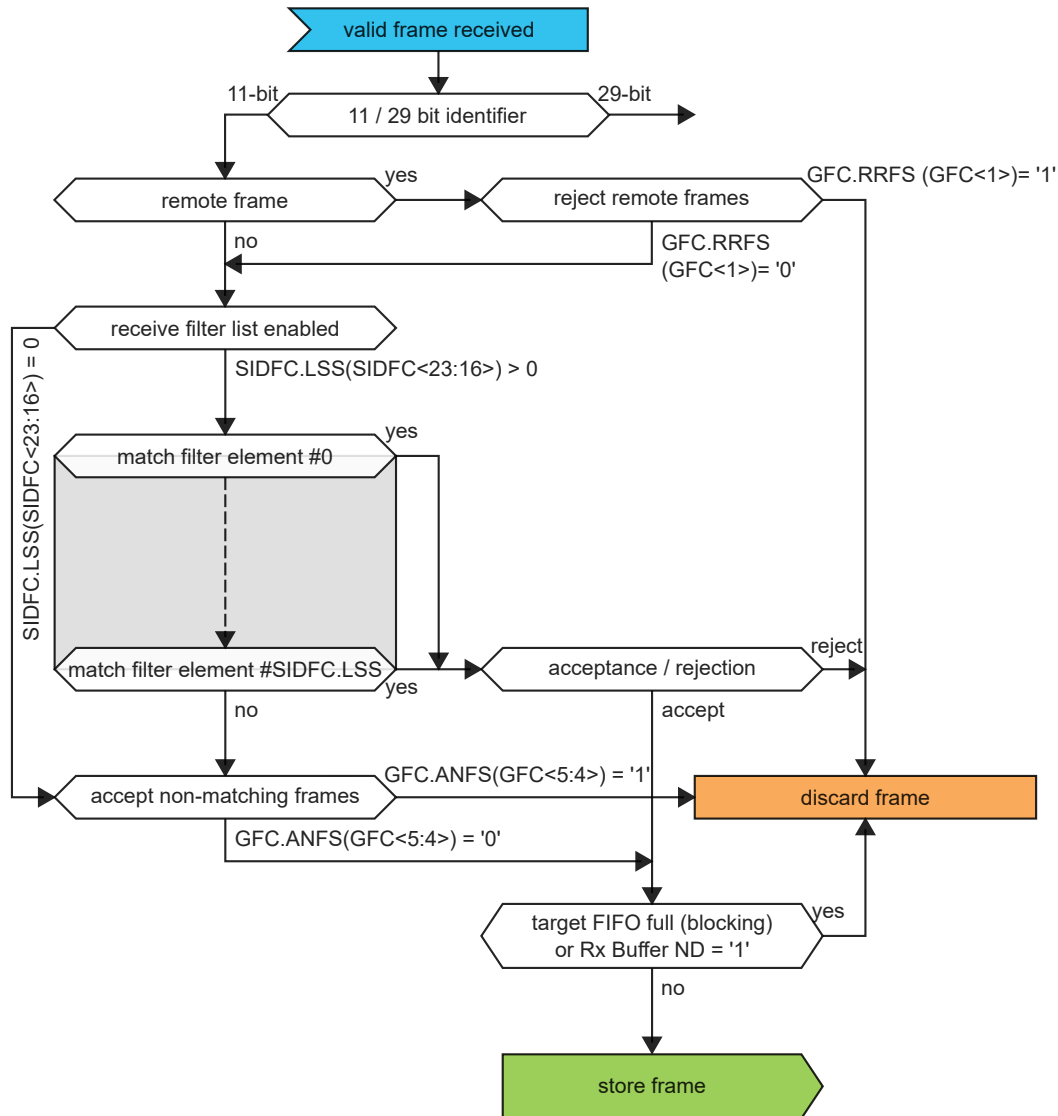
In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

Standard Message ID Filtering

The figure below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in [Standard Message ID Filter Element](#).

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

Figure 35-5. Standard Message ID Filtering



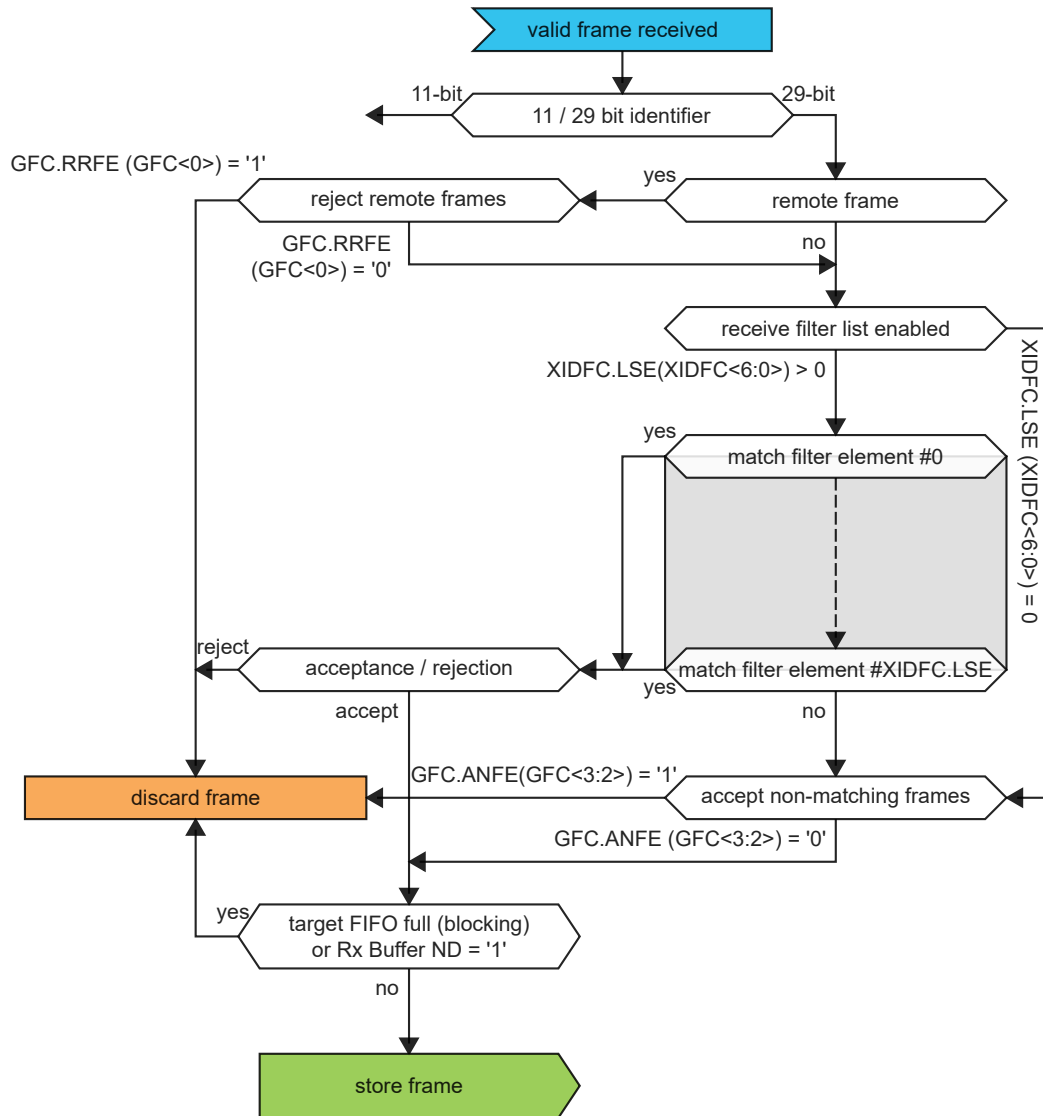
Extended Message ID Filtering

The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in *Extended Message ID Filter Element*.

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM is AND'ed with the received identifier before the filter list is executed.

Figure 35-6. Extended Message ID Filtering



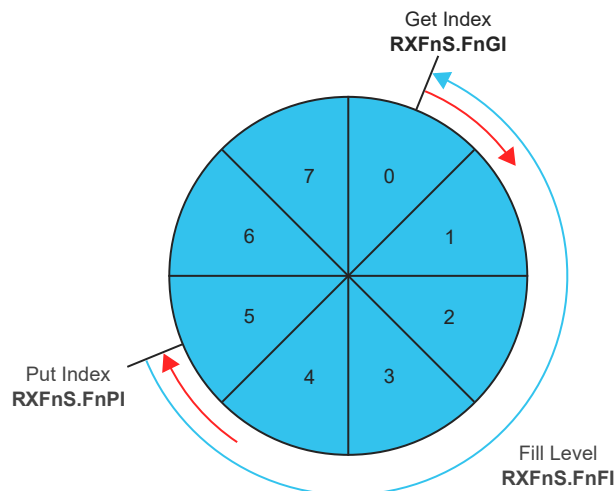
35.5.5.2 Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1 see *Acceptance Filtering*. The Rx FIFO element is described in *Rx Buffer and FIFO Element*.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by RXFnC.FnWM, interrupt flag IR.RFnW is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by RXFnS.FnF. In addition interrupt flag IR.RFnF is set, where n = 0 for Rx FIFO 0 buffer and n = 1 for Rx FIFO 1 buffer.

Figure 35-7. Rx FIFO Status (n =0 for Rx FIFO 0 buffer and n=1 for Rx FIFO 1 buffer)



When reading from an Rx FIFO, Rx FIFO Get Index $RXFnS.FnGI \cdot \text{FIFO Element Size}$ has to be added to the corresponding Rx FIFO start address $RXFnC.FnSA$.

Table 35-3. Rx Buffer / FIFO Element Size

$RXESC.RBDS[2:0]$ $RXESC.FnDS[2:0]$	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by $RXFnC.FnOM = '0'$. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached ($RXFnS.FnPI = RXFnS.FnGI$), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by $RXFnS.FnF = '1'$. In addition interrupt flag $IR.RFnF$ is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by $RXFnS.RFnL = '1'$. In addition interrupt flag $IR.RFnL$ is set.

Rx FIFO Overwrite Mode

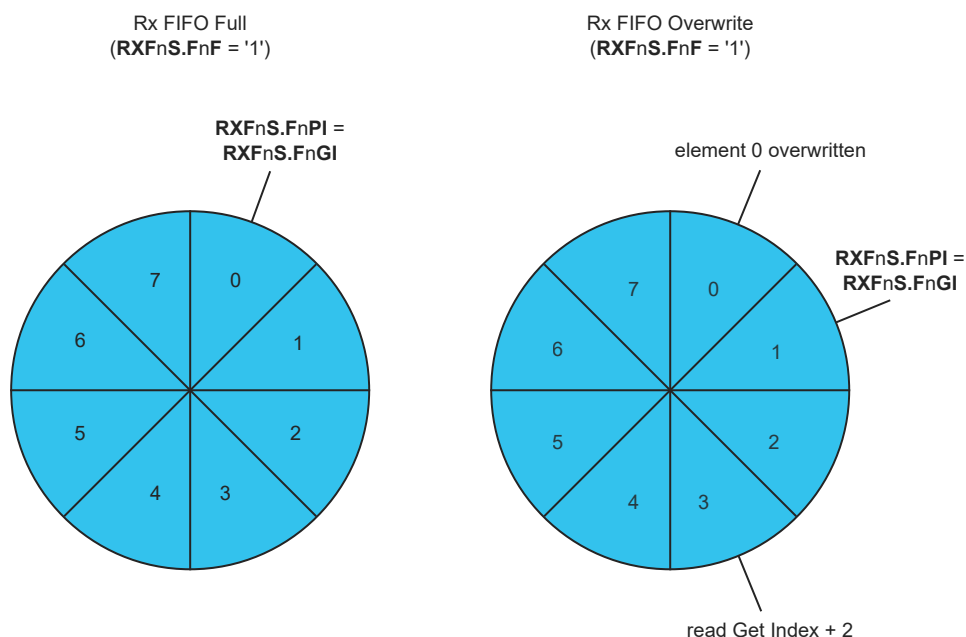
The Rx FIFO overwrite mode is configured by $RXFnC.FnOM = '1'$.

When an Rx FIFO full condition ($RXFnS.FnPI = RXFnS.FnGI$) is signaled by $RXFnS.FnF = '1'$, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signaled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might

happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. The figure below shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 35-8. Rx FIFO Overflow Handling (n =0 for Rx FIFO 0 buffer and n=1 for Rx FIFO 1 buffer)



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index $RXFnA.FnA$. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset ($RXFnS.FnF = '0'$).

35.5.5.3 Dedicated Rx Buffers

The CAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via $RXBC.RBSA$ bits ($RXBC <15:0>$).

For each Rx Buffer a Standard or Extended Message ID Filter Element with $SFEC / EFEC = "111"$ and $SFID2 / EFID2[10:9] = "00"$ has to be configured (see *Standard Message ID Filter Element* and *Extended Message ID Filter Element*).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag $IR.DRX$ bit ($IR <19>$) (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

Table 35-4. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] / EFID1[28:0]	SFID2[10:9] / EFID2[10:9]	SFID2[5:0] / EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register $NDAT1$, $NDAT2$ is set. As long as the New Data flag is set, the

respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

Rx Buffer Handling

- Reset interrupt flag IR.DRX bit (IR <19>)
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

35.5.5.4 Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see *Rx Buffer and FIFO Element*).

Advantage: Fixed start address for the DMA transfers (relative to RXBC.RBSA bits (RXBC <15:0>)), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = "111" have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the CAN while DMA request is activated. The behavior is similar to that of an Rx Buffers with its New Data flag set.

After the DMA has completed the DMA unit sets the DMA acknowledge. This resets DMA request. Now the CAN is prepared to receive the next set of debug messages.

Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to "111". In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see *Standard Message ID Filter Element* and *Extended Message ID Filter Element*). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor IR.DRX bit (IR <19>) are set. The reception of debug messages can be monitored via RXF1S.DMS bits (RXF1S <31:30>).

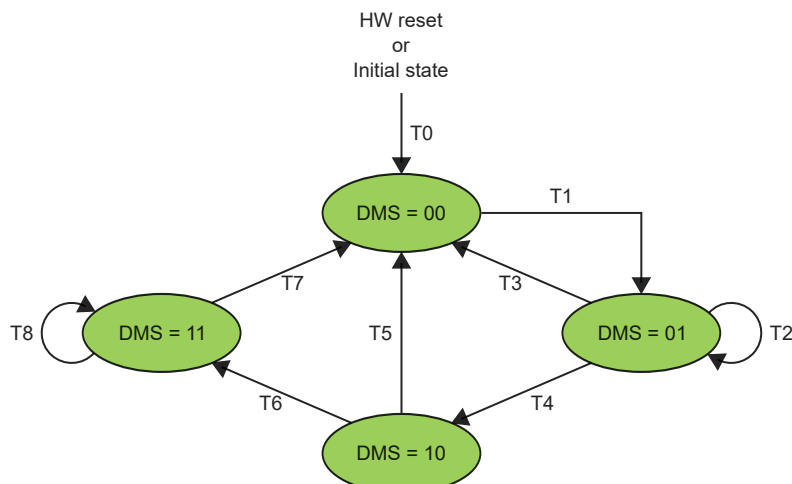
Table 35-5. Example Filter Configuration for Debug Messages

Filter Element	SFID1[10:0] / EFID1[28:0]	SFID2[10:9] / EFID2[10:9]	SFID2[5:0] / EFID2[5:0]
0	ID debug message A	01	11 1101
1	ID debug message B	10	11 1110
2	ID debug message C	11	11 1111

Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

Figure 35-9. Debug Message Handling State Machine



- T0: Reset DMA request output, enable reception of debug message A, B, and C
 T1: Reception of debug message A
 T2: Reception of debug message A
 T3: Reception of debug message C
 T4: Reception of debug message B
 T5: Reception of debug message A, B
 T6: Reception of debug message C
 T7: DMA transfer completed
 T8: Reception of debug message A, B, C (message rejected)

35.5.6 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in *Tx Buffer Element*. The table below describes the possible configurations for frame transmission.

Table 35-6. Possible Configurations for Frame Transmission

CCCR		Tx Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

35.5.6.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a

specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit CCCR.TXP bit (CCCR<14>). If the bit is set, the CAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP bit (CCCR<14>) = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

35.5.6.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (refer to table below). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0...31) • Element Size to the Tx Buffer Start Address TXBC.TBSA bit (TXBC <15:0>).

Table 35-7. Tx Buffer / FIFO / Queue Element Size

TXESC.TBDS[2:0]	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

35.5.6.3 Tx FIFO

Tx FIFO operation is configured by programming TXBC.TFQM bit (TXBC <30>) to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS.TFGI bit TXFQS (<12:8>). After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN calculates the Tx FIFO Free Level TXFQS.TFFL bits (TXFQS <5:0>) as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS.TFQPI bits TXFQS (<20:16>). An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF bit (TXFQS <21>) = '1') is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a '1' to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (refer to *Tx Buffer / FIFO / Queue Element Size*). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI bits (TXFQS <20:16>) (0...31).

- Element Size to the Tx Buffer Start Address TXBC.TBSA bits (TXBC <15:0>)

35.5.6.4 Tx Queue

Tx Queue operation is configured by programming TXBC.TFQM bit (TXBC <30>) to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TXFQS.TFQPI bits TXFQS (<20:16>). An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF bit (TXFQS <21>) = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been canceled.

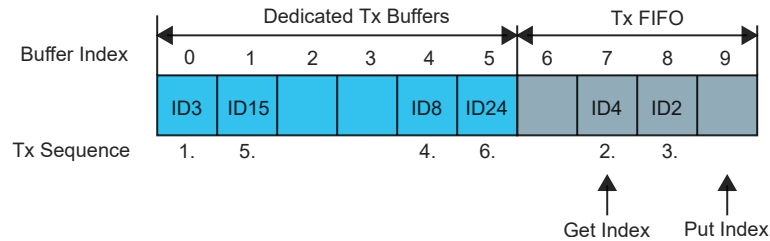
The application may use register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (refer to *Tx Buffer / FIFO / Queue Element Size*). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI bits TXFQS (<20:16>) (0...31). Element Size to the Tx Buffer Start Address TXBC.TBSA bits (TXBC <15:0>).

35.5.6.5 Mixed Dedicated Tx Buffers/Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO. The number of Dedicated Tx Buffers is configured by the TXBC.NDTB bits (TXBC <21:16>). The number of Tx Buffers assigned to the Tx FIFO is configured by the TXBC.TFQS bits (TXBC <29:24>). In case the TXBC.TFQS bits (TXBC <29:24>) is programmed to zero, only Dedicated Tx Buffers are used.

Figure 35-10. Example of mixed Configuration Dedicated Tx Buffers/Tx FIFO



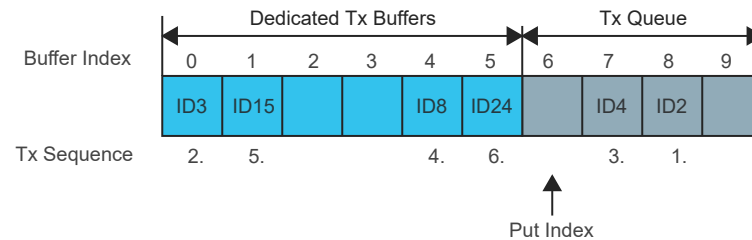
Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TXFQS.TFGI bits (TXFQS <12:8>)).
- Buffer with lowest Message ID gets highest priority and is transmitted next.

35.5.6.6 Mixed Dedicated Tx Buffers/Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue. The number of Dedicated Tx Buffers is configured by the TXBC.NDTB bits (TXBC <21:16>). The number of Tx Queue Buffers is configured by the TXBC.TFQS bits (TXBC <29:24>). In case the TXBC.TFQS bits (TXBC <29:24>) is programmed to zero, only Dedicated Tx Buffers are used.

Figure 35-11. Example of mixed Configuration Dedicated Tx Buffers/Tx Queue



Tx prioritization:

- Scan all Tx Buffers with activated transmission request.
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next.

35.5.6.7 Transmit Cancellation

The CAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR-based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a '1' to the corresponding bit position (equal to number of Tx Buffer) of the TXBCR register. Transmit cancellation is not intended for the Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of the TXBCF register to '1'.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding TXBTO and TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding TXBCF bit is set.

Note: In case a pending transmission is canceled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

35.5.6.8 Tx Event Handling

To support Tx event handling, the CAN has implemented a Tx Event FIFO. After the CAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in *Tx Event FIFO Element*.

When a Tx Event FIFO full condition is signaled by the IR.TEFF bit (IR <14>), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and the Interrupt Flag IR.TEFL bit (IR <15>) is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the TXEFC.EFWM bit (TXEFC <29:24>), the Interrupt Flag IR.TEFW bit (IR <13>) is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS.EFGI bit (TXEFS <12:8>) has to be added to the Tx Event FIFO start address TXEFC.EFSA bit (TXEFC <15:0>).

35.5.7 FIFO Acknowledge Handling

The Get Indexes of Rx FIFO 0, Rx FIFO 1 and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to the *RXF0A*, *RXF1A* and *TXEFA*). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the CAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a high-priority message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN does not check for erroneous values.

35.5.8 Message RAM

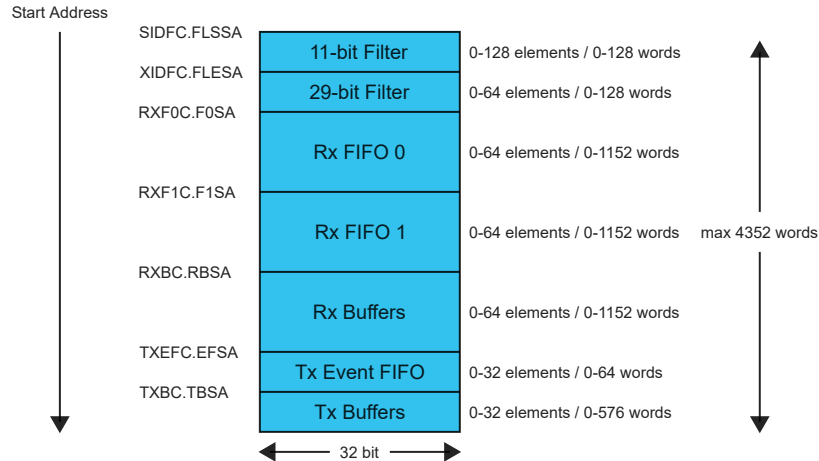
For storage of Rx/Tx messages and for storage of the filter configuration this module uses system RAM and the base address can be set by the CTRLB.OFFSET bits (CTRLB <23:16>).

35.5.8.1 Message RAM Configuration

The Message RAM has a width of 32 bits. The CAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in the figure below, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO 0, Rx FIFO 1, Rx Buffers, and Tx Buffers through the RXESC.F0DS bits (RXESC <2:0>), RXESC.F1DS bits (RXESC <6:4>), RXESC.RBDS bits (RXESC <10:8>), and TXESC.TBDS bits (TXESC <2:0>).

Figure 35-12. Message RAM Configuration



When the CAN addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses (i.e., only bits 15 to 2 are evaluated and the two LSBs are ignored).



WARNING The CAN does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.

35.5.8.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer and FIFO element is shown in the table below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field through the RXESC register.

Table 35-8. Rx Buffer and FIFO Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	E S I	X T D	R T R	ID[28:0]																												
R1	A N M F	FIDX[6:0]									F D F	B R S	DLC[3:0]			RXTS[15:0]																
R2	DB3[7:0]							DB2[7:0]							DB1[7:0]							DB0[7:0]										
R3	DB7[7:0]							DB6[7:0]							DB5[7:0]							DB4[7:0]										
...	...							...							...							...										
Rn	DBm[7:0]							DBm-1[7:0]							DBm-2[7:0]							DBm-3[7:0]										

R0 Bit 31 - ESI: Error State Indicator

0 : Transmitting node is error active.

1 : Transmitting node is error passive.

R0 Bit 30 - XTD: Extended Identifier

Signals to the Host whether the received frame has a standard or extended identifier.

0 : 11-bit standard identifier.

1 : 29-bit extended identifier.

R0 Bit 29 - RTR: Remote Transmission Request

Signals to the Host whether the received frame is a data frame or a remote frame.

0 : Received frame is a data frame.

1 : Received frame is a remote frame.

Note: There are no remote frames in CAN FD format. If a CAN FD frame was received (EDL = '1'), bit RTR reflects the state of the reserved bit r1.

R0 Bits 28:0 - ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

R1 Bit 31 - ANMF: Accepted Non-matching Frame

Acceptance of non-matching frames may be enabled via GFC.ANFS bits (GFC <5:4>) and GFC.ANFE bits (GFC <3:2>).

0 : Received frame matching filter index FIDX.

1 : Received frame did not match any Rx filter element.

R1 Bits 30:24 - FIDX[6:0]: Filter Index

0-127 : Index of matching Rx acceptance filter element (invalid if ANMF = '1').

Note: Range is 0 to SIDFC.LSS bits (SIDFC <23:16>) -1 for standard and 0 to XIDFC.LSE bits (XIDFC <22:16>) -1 for extended.

R1 Bits 23:22 - Reserved

R1 Bit 21 - FDF: FD Format

0 : Standard frame format.

1 : CAN FD frame format (new DLC-coding and CRC).

R1 Bit 20 - BRS: Bit Rate Search

0 : Frame received without bit rate switching.

1 : Frame received with bit rate switching.

R1 Bits 19:16 - DLC[3:0]: Data Length Code

0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

9-15 : CAN: received frame has 8 data bytes.

9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

R1 Bits 15:0 - RXTS[15:0]: Rx Timestamp

Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP bit (TSCC <19:16>).

R2 Bits 31:24 - DB3[7:0]: Data Byte 3

R2 Bits 23:16 - DB2[7:0]: Data Byte 2

R2 Bits 15:8 - DB1[7:0]: Data Byte 1

R2 Bits 7:0 - DB0[7:0]: Data Byte 0

R3 Bits 31:24 - DB7[7:0]: Data Byte 7

R3 Bits 23:16 - DB6[7:0]: Data Byte 6

R3 Bits 15:8 - DB5[7:0]: Data Byte 5

R3 Bits 7:0 - DB4[7:0]: Data Byte 4

...

Rn Bits 31:24 - DBm[7:0]: Data Byte m

Rn Bits 23:16 - DBm-1[7:0]: Data Byte m-1

Rn Bits 15:8 - DBm-2[7:0]: Data Byte m-2

Rn Bits 7:0 - DBm-3[7:0]: Data Byte m-3



WARNING Depending on the configuration of RXESC, between two and sixteen 32-bit words (Rn = 3 ... 17) are used for storage of a CAN message's data field.

35.5.8.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS bits (TXBC <29:24>) and TXBC.NDTB bits (TXBC <21:16>). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 35-9. Tx Buffer Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	E S I	X T D	R T R	ID[28:0]																												
T1	MM[7:0]								E F C		F D F	B R S	DLC[3:0]																			
T2	DB3[7:0]								DB2[7:0]				DB1[7:0]				DB0[7:0]															
T3	DB7[7:0]								DB6[7:0]				DB5[7:0]				DB4[7:0]															
...	...								...				...				...															
Tn	DBm[7:0]								DBm-1[7:0]				DBm-2[7:0]				DBm-3[7:0]															

T0 Bit 31 - ESI: Error State Indicator

0 : ESI bit in CAN FD format depends only on error passive flag.

1 : ESI bit in CAN FD format transmitted recessive.

Note: The ESI bit of the transmit buffer is OR'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.

T0 Bit 30 - XTD: Extended Identifier

0 : 11-bit standard identifier.

1 : 29-bit extended identifier.

T0 Bit 29 - RTR: Remote Transmission Request

0 : Transmit data frame.

1 : Transmit remote frame.

Note: When RTR = '1', the CAN transmits a remote frame according to ISO 11898-1, even if CCCR.CME enables the transmission in CAN FD format.

T0 Bits 28:0 - ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

T1 Bits 31:24 - MM[7:0]: Message Marker

Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

T1 Bit 23 - EFC: Event FIFO Control

0 : Don't store Tx events.

1 : Store Tx events.

T1 Bit 22 - Reserved

TR1 Bit 21 - FDF: FD Format

0 : Frame transmitted in Classic CAN format.

1 : Frame transmitted in CAN FD format.

T1 Bit 20 - BRS: Bit Rate Search

0 : CAN FD frames transmitted without bit rate switching.

1 : CAN FD frames transmitted with bit rate switching.

Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR.FDOE bit (CCCR <8>) = '1'. Bit BRS is only evaluated when in addition CCCR.BRSE bit (CCCR <9>) = '1'.

T1 Bits 19:16 - DLC[3:0]: Data Length Code

0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

9-15 : CAN: received frame has 8 data bytes.

9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

T1 Bits 15:0 - Reserved

T2 Bits 31:24 - DB3[7:0]: Data Byte 3

T2 Bits 23:16 - DB2[7:0]: Data Byte 2

T2 Bits 15:8 - DB1[7:0]: Data Byte 1

T2 Bits 7:0 - DB0[7:0]: Data Byte 0

T3 Bits 31:24 - DB7[7:0]: Data Byte 7

T3 Bits 23:16 - DB6[7:0]: Data Byte 6

T3 Bits 15:8 - DB5[7:0]: Data Byte 5

T3 Bits 7:0 - DB4[7:0]: Data Byte 4

...

Tn Bits 31:24 - DBm[7:0]: Data Byte m

Tn Bits 23:16 - DBm-1[7:0]: Data Byte m-1

Tn Bits 15:8 - DBm-2[7:0]: Data Byte m-2

Tn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

Note: Depending on the configuration of TXESC, between two and sixteen 32-bit words (Tn = 3 ... 17) are used for storage of a CAN message's data field.

35.5.8.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

Table 35-10. Tx Event FIFO Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E0	ESI	XTR		ID[28:0]																												
E1	MM[7:0]								ET [1:0]	F D F	B R S	DLC[3:0]				TXTS[15:0]																

E0 Bit 31 - ESI: Error State Indicator

- 0 : Transmitting node is error active.
- 1 : Transmitting node is error passive.

E0 Bit 30 - XTD: Extended Identifier

- 0 : 11-bit standard identifier.
- 1 : 29-bit extended identifier.

E0 Bit 29 - RTR: Remote Transmission Request

- 0 : Received frame is a data frame.
- 1 : Received frame is a remote frame.

E0 Bits 28:0 - ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

E1 Bits 31:24 - MM[7:0]: Message Marker

Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.

E1 Bits 23:22 - ET[1:0]: Event Type

This field defines the event type.

Table 35-11. Event Type

Value	Name	Description
0x0 or 0x3	RES	Reserved
0x1	TXE	Tx event
0x2	TXC	Transmission in spite of cancellation (always set for transmission in DAR mode)

E1 Bit 21 - FDF: FD Format

- 0 : Standard frame format.
- 1 : CAN FD frame format (new DLC-coding and CRC).

E1 Bit 20 - BRS: Bit Rate Search

- 0 : Frame received without bit rate switching.

1 : Frame received with bit rate switching.

E1 Bits 19:16 - DLC[3:0]: Data Length Code

0-8 : CAN + CAN FD: received frame has 0-8 data bytes.

9-15 : CAN: received frame has 8 data bytes.

9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.

E1 Bits 15:0 - TXTS[15:0]: Tx Timestamp

Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP bit (TSCC <19:16>).

35.5.8.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address SIDFC.FLSSA bits (SIDFC <15:0>) plus the index of the filter element (0 ... 127).

Table 35-12. Standard Message ID Filter Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S0	SFT [1:0]		SFEC [2:0]		SFID1[10:0]														SFID2[10:0]													

Bits 31:30 - SFT[1:0]: Standard Filter Type

This field defines the standard filter type.

Table 35-13. Standard Filter Type

Value	Name	Description
0x0	RANGE	Range filter from SFID1 to SFID2 (SFID2 >= SFID1)
0x1	DUAL	Dual ID filter for SFID1 or SFID2
0x2	CLASSIC	Classic filter: SFID1 = filter, SFID2 = mask
0x3	RES	Reserved

Bits 29:27 - SFEC[2:0]: Standard Filter Element Configuration

All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = "100", "101", or "110" a match sets interrupt flag IR.HPM (IR <8>) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Table 35-14. Standard Filter Element Configuration

Value	Name	Description
0x0	DISABLE	Disable filter element
0x1	STF0M	Store in Rx FIFO 0 if filter matches
0x2	STF1M	Store in Rx FIFO 1 if filter matches
0x3	REJECT	Reject ID if filter matches
0x4	PRIORITY	Set priority if filter matches.
0x5	PRIF0M	Set priority and store in FIFO 0 if filter matches.
0x6	PRIF1M	Set priority and store in FIFO 1 if filter matches.
0x7	STRXBUF	Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored.

Bits 26:16 - SFID1[10:0]: Standard Filter ID 1

First ID of standard ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.

Bits 15:11 - Reserved

Bits 10:0 - SFID2[10:0]: Standard Filter ID 2

This bit field has a different meaning depending on the configuration of SFEC.

1. SFEC = "001" ... "110": Second ID of standard ID filter element.
2. SFEC = "111": Filter for Rx Buffers or for debug messages.

SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

00 = Store message into an Rx Buffer

01 = Debug Message A

10 = Debug Message B

11 = Debug Message C

SFID2[8:6] is used to control the filter event pins at the Extension Interface. A '1' at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CANx_APB period in case the filter matches.

SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA bits (RXBC <15:0>) for storage of a matching message.

35.5.8.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFC.FLESA bits (XIDFC <15:0>) plus two times the index of the filter element (0...63).

Table 35-15. Extended Message ID Filter Element

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0	EFEC [2:0]			EFID1[28:0]																												
F1	EFT [1:0]			EFID2[28:0]																												

F0 Bits 31:29 - EFEC[2:0]: Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = "100", "101", or "110" a match sets interrupt flag IR.HPM (IR <8>) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

Table 35-16. Extended Filter Element Configuration

Value	Name	Description
0x0	DISABLE	Disable filter element.
0x1	STF0M	Store in Rx FIFO 0 if filter matches.
0x2	STF1M	Store in Rx FIFO 1 if filter matches.
0x3	REJECT	Reject ID if filter matches.
0x4	PRIORITY	Set priority if filter matches.
0x5	PRIF0M	Set priority and store in FIFO 0 if filter matches.
0x6	PRIF1M	Set priority and store in FIFO 1 if filter matches.
0x7	STRXBUF	Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored.

F0 Bits 28:0 - EFID1[28:0]: Extended Filter ID 1

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of a extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism is used.

F1 Bits 31:30 - EFT[1:0]: Extended Filter Type

This field defines the extended filter type.

Table 35-17. Extended Filter Type

Value	Name	Description
0x0	RANGEM	Range filter from EFID1 to EFID2 (EFID2 >= EFID1).
0x1	DUAL	Dual ID filter for EFID1 or EFID2.
0x2	CLASSIC	Classic filter: EFID1 = filter, EFID2 = mask.
0x3	RANGE	Range filter from EFID1 to EFID2 (EFID2 >= EFID1), XIDAM mask not applied.

F1 Bits 28:0 - EFID2[28:0]: Extended Filter ID 2

This bit field has a different meaning depending on the configuration of EFEC.

- 1) EFEC = "001" ... "110" Second ID of standard ID filter element.
- 2) EFEC = "111" Filter for Rx Buffers or for debug messages.

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

00 = Store message into an Rx Buffer

01 = Debug Message A

10 = Debug Message B

11 = Debug Message C

EFID2[8:6] is used to control the filter event pins at the Extension Interface. A '1' at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CANx_AHB period in case the filter matches.

EFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA bits (RXBC <15:0>) for storage of a matching message.

35.5.9 Interrupts

The CAN has the following interrupt sources, which can be assigned to 2 interrupt lines:

- Access to Reserved Address
- Protocol Errors (Data Phase / Arbitration Phase)
- Watchdog Interrupt
- "bus off" Status
- Error Warning & Passive
- Error Logging Overflow
- Message stored to Dedicated Rx Buffer
- Timeout Occurred
- Message RAM Access Failure
- Timestamp Wraparound
- Tx Event FIFO statuses (Element Lost / Full / Watermark Reached / New Entry)
- Tx FIFO Empty
- Transmission Cancellation Finished

- Timestamp Completed
- High Priority Message
- Rx FIFO 1 Statuses (Message Lost / Full / Watermark Reached / New Message)
- Rx FIFO 0 Statuses (Message Lost / Full / Watermark Reached / New Message)

Each interrupt source has an interrupt flag associated with it. The interrupt flag register (IR) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing '1' or disabled by writing '0' to the corresponding bit in the interrupt enable register (IE). Each interrupt flag can be assigned to one of two interrupt service lines.

An interrupt request is generated when an interrupt flag is set, the corresponding interrupt enable is set, and the corresponding service line enable assigned to the interrupt is set. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the service line is disabled, or the CAN is reset. Refer to the IR register for details on how to clear interrupt flags. The interrupt request lines are connected to the Nested Vector Interrupt Controller (NVIC). All interrupt requests from the peripheral are sent to the *Nested vector Interrupt Controller NVIC*. Using the CAN FD interrupt requires the interrupt controller to be configured first. The user must read the IR register to determine which interrupt condition is present.

The CAN also has two other interrupt sources:

- AHB Bus Error (BERR)
- Debug Message Reception Interrupt (DBG)

These two interrupt lines have dedicated bit in the INTFLAG register. The user must read the INTFLAG register to determine which interrupt condition is present.

The interrupt source can be enabled by writing a one to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). The interrupt request is generated when the interrupt flag is set, and the interrupt is enabled. The interrupt request remains active until the interrupt flag source of interrupt is cleared, or when the interrupt source is disabled. Refer to INTFLAG register for details on how to clear interrupt flags.

The interrupt flags are cleared in the case of a hardware reset condition detection.

Note: Interrupts must be globally enabled for interrupt requests to be generated.

35.5.10 Sleep Mode Operation

The CAN module can be configured to operate in any idle sleep mode if both AHB and GCLK clocks are available. The CAN interrupts can be used to wake up the device from sleep mode.

The CAN module has its own low power mode that may be used at any time without disabling this module. This is performed by writing one to the Clock Stop Request bit in the CC Control register (CCCR.CSR bit (CCCR <4>) = 1). Once all pending transactions are completed and the idle bus state is detected, the CAN will automatically set the Clock Stop Acknowledge bit (CCCR.CSA bit (CCCR <3>) = 1). The CAN then reverts back to its initial state (CCCR.INIT bit (CCCR <0>) = 1), blocking further transfers.

To exit low power mode, CCCR.CSR bit (CCCR<4>) in CCCR register must be written to 0. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT bit (CCCR <0>).

After reset, the GCLK_CANx and CLK_CANx_AHB clocks are not requested, except for each APB bus access. However, after the CAN initialization, both GCLK_CANx and CLK_CANx_AHB clocks are requested as long as Clock Stop Request bit in the CC Control register is cleared (CCCR.CSR (CCCR<4>) = 0), and stopped when Clock Stop Request bit in the CC Control register is set (CCCR.CSR (CCCR<4>) = 1) and the Clock Stop Acknowledge bit is set (CCCR.CSA (CCCR<3>) = 1). To limit the wake-up time latency, the CAN clock sources must be enabled in continuous mode (ONDEMAND of respective oscillator must be set zero). For further details, refer to the *Clock System* chapter.

In Standby sleep mode, when the RUNSTDBY bit is set (CTRLA.RUNSTDBY=1), the CAN IP is allowed to continue to run its operation.

When the RUNSTDBY bit is cleared (CTRLA.RUNSTDBY=0) and the Standby sleep mode is detected, a hardware mechanism will request the CAN peripheral to stop its operation, and enables the CAN low power mode automatically. The CCCR.CSR bit is automatically set to one as long as the stop request operation is active.

When all pending transmission requests have completed, the CAN waits until bus idle state is detected. Then the CAN sets the CCCR.INIT to one to prevent any further CAN transfers. The hardware stop request is acknowledged and the CCCR.CSA bit is set one. The clocks requests are switched off.

When the system exits the Standby sleep mode, CSR bit in CCCR register must be written to 0. Afterward, the application can restart CAN communication by resetting bit CCCR.INIT.

Note: In case of a heavily disturbed CAN bus, it may happen that an idle state is never reached and CCCR.INIT is therefore not set by the CAN. This situation can be detected by polling PSR.ACT. In case there is a risk the CAN does not enter idle state, it is recommended to use the software procedure to put the CAN in low power mode, before the system goes to Standby sleep mode. In this way, the software can write CCCR.INIT = '1', which stops CAN communication immediately, regardless whether there is a transmission or reception ongoing or not.

35.5.11 Synchronization

Due to the asynchronicity between the main clock domain (CLK_CANx_AHB) and the peripheral clock domain (GCLK_CANx) some registers are synchronized when written. When a write-synchronized register is written, the read back value will not be updated until the register has completed synchronization.

The CLK_CANx_AHB clock must be faster or equal to the GCLK_CANx clock.

The following bits and registers are write-synchronized:

One Initialization bit in CC Control register (CCCR.INIT bit (CCCR <0>)).

35.6 Peripheral Dependencies

Table 35-18. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
CAN0	0x4402E000	35 : INT0 36 : INT1 37 : BERR/DBG	AHB: CLKMSK[0] MASK23	PCHCTRL[6]	17 STATUS0 PERID17			PD_CORE_SW
CAN1	0x44030000	38 : INT0 39 : INT1 40 : BERR/DBG	AHB: CLKMSK[0] MASK24	PCHCTRL[7]	18 STATUS0 PERID18			PD_CORE_SW

35.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0		RUNSTDBY				PRIV		SWRST
0x04	CTRLB	31:24								
		23:16	OFFSET[7:0]							
		15:8								
		7:0								
0x08 ... 0x27	Reserved									
0x28	INTENCLR	31:24								
		23:16								
		15:8								
		7:0							BERR	DBG
0x2C	INTENSET	31:24								
		23:16								
		15:8								
		7:0							BERR	DBG
0x30	INTFLAG	31:24								
		23:16								
		15:8								
		7:0							BERR	DBG
0x34	INTFLAGSET	31:24								
		23:16								
		15:8								
		7:0							BERR	DBG
0x38 ... 0x5B	Reserved									
0x5C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0								SWRST
0x60 ... 0xDF	Reserved									
0xE0	WPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPLCK	WPEN
0xE4 ... 0xFF	Reserved									
0x0100	CREL_MCAN	31:24	REL[3:0]				STEP[3:0]			
		23:16	SUBSTEP[3:0]				YEAR[3:0]			
		15:8					MON[7:0]			
		7:0					DAY[7:0]			
0x0104	ENDN	31:24					ETV[31:24]			
		23:16					ETV[23:16]			
		15:8					ETV[15:8]			
		7:0					ETV[7:0]			
0x0108	CUST	31:24					CUST[31:24]			
		23:16					CUST[23:16]			
		15:8					CUST[15:8]			
		7:0					CUST[7:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x010C	DBTP	31:24								
		23:16	TDC				DBRP[4:0]			
		15:8					DTSEG1[4:0]			
		7:0	DTSEG2[3:0]				DSJW[3:0]			
0x0110	TEST	31:24								
		23:16								
		15:8								
		7:0	RX	TX[1:0]		LBCK				
0x0114	RWD	31:24								
		23:16								
		15:8					WDV[7:0]			
		7:0					WDC[7:0]			
0x0118	CCCR	31:24								
		23:16								
		15:8		TXP	EFBI	PXHD	WMM	UTSU	BRSE	FDOE
		7:0	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
0x011C	NBTP	31:24				NSJW[6:0]				NBRP[8]
		23:16					NBRP[7:0]			
		15:8					NTSEG1[7:0]			
		7:0					NTSEG2[6:0]			
0x0120	TSCC	31:24								
		23:16					TCP[3:0]			
		15:8								
		7:0							TSS[1:0]	
0x0124	TSCV	31:24								
		23:16								
		15:8				TSC[15:8]				
		7:0				TSC[7:0]				
0x0128	TOCC	31:24					TOP[15:8]			
		23:16					TOP[7:0]			
		15:8								
		7:0						TOS[1:0]		ETOC
0x012C	TOCV	31:24								
		23:16								
		15:8				TOC[15:8]				
		7:0				TOC[7:0]				
0x0130 ... 0x013F	Reserved									
0x0140	ECR	31:24								
		23:16				CEL[7:0]				
		15:8	RP				REC[6:0]			
		7:0					TEC[7:0]			
0x0144	PSR	31:24								
		23:16				TDCV[6:0]				
		15:8		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
		7:0	BO	EW	EP	ACT[1:0]		LEC[2:0]		
0x0148	TDCR	31:24								
		23:16								
		15:8				TDCO[6:0]				
		7:0				TDCF[6:0]				
0x014C ... 0x014F	Reserved									
0x0150	IR	31:24			ARA	PED	PEA	WDI	BO	EW
		23:16	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
		15:8	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
		7:0	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0	
0x0154	IE	31:24			ARAE	PEDE	PEAE	WDIE	BOE	EWE	
		23:16	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE	
		15:8	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	
		7:0	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RFONE	
0x0158	ILS	31:24			ARAL	PEDL	PEAL	WDIL	BOL	EWL	
		23:16	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL	
		15:8	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	
		7:0	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RFONL	
0x015C	ILE	31:24									
		23:16									
		15:8									
		7:0							EINT1	EINT0	
0x0160 ... 0x017F	Reserved										
0x0180	GFC	31:24									
		23:16									
		15:8									
		7:0			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE	
0x0184	SIDFC	31:24									
		23:16	LSS[7:0]								
		15:8	FLSSA[13:6]								
		7:0	FLSSA[5:0]								
0x0188	XIDFC	31:24									
		23:16		LSE[6:0]							
		15:8	FLESA[13:6]								
		7:0	FLESA[5:0]								
0x018C ... 0x018F	Reserved										
0x0190	XIDAM	31:24				EIDM[28:24]					
		23:16	EIDM[23:16]								
		15:8	EIDM[15:8]								
		7:0	EIDM[7:0]								
0x0194	HPMS	31:24									
		23:16									
		15:8	FLST	FIDX[6:0]							
		7:0	MSI[1:0]		BIDX[5:0]						
0x0198	NDAT1	31:24	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	
		23:16	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	
		15:8	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	
		7:0	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0	
0x019C	NDAT2	31:24	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	
		23:16	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	
		15:8	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	
		7:0	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32	
0x01A0	RXFOC	31:24	F0OM	F0WM[6:0]							
		23:16		F0S[6:0]							
		15:8	F0SA[13:6]								
		7:0	F0SA[5:0]								
0x01A4	RXF0S	31:24							RFOL	F0F	
		23:16			F0PI[5:0]						
		15:8			F0GI[5:0]						
		7:0		F0FL[6:0]							
0x01A8	RXFOA	31:24									
		23:16									
		15:8									
		7:0			F0AI[5:0]						

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01AC	RXBC	31:24								
		23:16								
		15:8	RBSA[13:6]							
		7:0	RBSA[5:0]							
0x01B0	RXF1C	31:24	F1OM					F1WM[6:0]		
		23:16						F1S[6:0]		
		15:8						F1SA[13:6]		
		7:0						F1SA[5:0]		
0x01B4	RXF1S	31:24	DMS[1:0]						RF1L	F1F
		23:16						F1PI[5:0]		
		15:8						F1GI[5:0]		
		7:0						F1FL[6:0]		
0x01B8	RXF1A	31:24								
		23:16								
		15:8								
		7:0						F1AI[5:0]		
0x01BC	RXESC	31:24								
		23:16								
		15:8							RBDS[2:0]	
		7:0			F1DS[2:0]				F0DS[2:0]	
0x01C0	TXBC	31:24		TFQM				TFQS[5:0]		
		23:16						NDTB[5:0]		
		15:8						TBSA[13:6]		
		7:0						TBSA[5:0]		
0x01C4	TXFQS	31:24								
		23:16			TFQF			TFQPI[4:0]		
		15:8						TFGI[4:0]		
		7:0						TFFL[5:0]		
0x01C8	TXESC	31:24								
		23:16								
		15:8								
		7:0						TBDS[2:0]		
0x01CC	TXBRP	31:24	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
		23:16	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
		15:8	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
		7:0	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
0x01D0	TXBAR	31:24	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
		23:16	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
		15:8	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
		7:0	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
0x01D4	TXBCR	31:24	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
		23:16	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
		15:8	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
		7:0	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
0x01D8	TXBTO	31:24	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
		23:16	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
		15:8	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
		7:0	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
0x01DC	TXBCF	31:24	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
		23:16	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
		15:8	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
		7:0	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
0x01E0	TXBTIE	31:24	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
		23:16	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
		15:8	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
		7:0	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
0x01E4	TXBCIE	31:24	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
		23:16	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
		15:8	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
		7:0	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01E8 ... 0x01EF	Reserved									
0x01F0	TXEFC	31:24						EFWM[5:0]		
		23:16						EFS[5:0]		
		15:8					EFSA[13:6]			
		7:0			EFSA[5:0]				TEFL	EFF
0x01F4	TXEFS	31:24								
		23:16					EFPI[4:0]			
		15:8					EFGI[4:0]			
		7:0					EFFL[5:0]			
0x01F8	TXEFA	31:24								
		23:16								
		15:8								
		7:0					EFAI[4:0]			
0x01FC ... 0x025F	Reserved									
0x0260	CREL_TSU	31:24			REL[3:0]				STEP[3:0]	
		23:16			SUBSTEP[3:0]				YEAR[3:0]	
		15:8					MON[7:0]			
		7:0					DAY[7:0]			
0x0264	TSCFG	31:24								
		23:16								
		15:8					TBPRE[7:0]			
		7:0						SCP	TBCS	TSUE
0x0268	TSS1	31:24					TSN[15:8]			
		23:16					TSN[7:0]			
		15:8					TSL[15:8]			
		7:0					TSL[7:0]			
0x026C	TSS2	31:24								
		23:16								
		15:8		ITBG[1:0]	NTSG[1:0]					
		7:0						TSP[3:0]		
0x0270	TS0	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0274	TS1	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0278	TS2	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x027C	TS3	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0280	TS4	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0284	TS5	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x0288	TS6	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x028C	TS7	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0290	TS8	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0294	TS9	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x0298	TS10	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x029C	TS11	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x02A0	TS12	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x02A4	TS13	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x02A8	TS14	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x02AC	TS15	31:24					TS[31:24]			
		23:16					TS[23:16]			
		15:8					TS[15:8]			
		7:0					TS[7:0]			
0x02B0	ATB	31:24					TB[31:24]			
		23:16					TB[23:16]			
		15:8					TB[15:8]			
		7:0					TB[7:0]			

35.7.1 Control A Register

Name: CTRLA
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 35-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
		RUNSTDBY				PRIV		SWRST
Access		R/W				R/W		R/HC/S
Reset		0				0		0

Bit 6 – RUNSTDBY Run during standby

Value	Name	Description
0	DISABLE	The module is internally disabled and stops the operation when all bus transactions are completed.
1	ENABLE	The module continues the normal operation in Standby sleep mode.

Bit 2 – PRIV Privileged Access

Bit 0 – SWRST Software Reset

Writing a '1' to this bit will reset all APB registers, including the M_CAN and TSU registers. The bit stays high until reset completes. Setting this bit also sets the SYNCBUSY.SWRST to 1. SYNCBUSY.SWRST stays 1 until reset sequence completes.

Value	Name	Description
0	NONE	No Action.
1	RESET	Reset all APB registers.

35.7.2 Control B Register

Name: CTRLB
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 35-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	OFFSET[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 23:16 – OFFSET[7:0] Message RAM Base Address Offset.

This bitfield value represents the 8 bits offset of the memory base address (bits [23:16]). The base address is calculated using the following formula:

Base_Address = 0x20 << 24 + OFFSET << 16.

Note: It is recommended to write this register only when the Change Configuration bit in the CC Control register is set (CCCR.CCE = 1).

35.7.3 Interrupt Enable Clear register

Name: INTENCLR
Offset: 0x028
Reset: 0x00000000
Property: R/K

Table 35-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							BERR	DBG
Access							R/K	R/K
Reset							0	0

Bit 1 – BERR Bus Error Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Bus Error Interrupt Enable bit, which disables the Bus Error interrupt.

Value	Description
0	The Bus Error Interrupt is disabled.
1	The Bus Error Interrupt is enabled, and an interrupt request is generated when the Bus Error Message Interrupt Flag is set.

Bit 0 – DBG Debug Message Interrupt Disable

Writing '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Message Interrupt Enable bit, which disables the Debug Message Interrupt.

Value	Description
0	The Debug Message Interrupt is disabled.
1	The Debug Message Interrupt is enabled, and an interrupt request is generated when the Debug Message Interrupt Flag is set.

35.7.4 Interrupt Enable Set register

Name: INTENSET
Offset: 0x02C
Reset: 0x00000000
Property: R/S

Table 35-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							BERR	DBG
Access							R/S	R/S
Reset							0	0

Bit 1 – BERR Bus Error Interrupt Enable

Writing '0' to this bit has no effect.

Writing '1' to this bit will clear the Bus Error Interrupt Enable bit, which disables the Bus Error interrupt.

Value	Description
0	The Bus Error Interrupt is disabled.
1	The Bus Error Interrupt is enabled, and an interrupt request is generated when the Bus Error Message Interrupt Flag is set.

Bit 0 – DBG Debug Message Interrupt Enable

Writing '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Message Interrupt Enable bit, which disables the Debug Message Interrupt.

Value	Description
0	The Debug Message Interrupt is disabled.
1	The Debug Message Interrupt is enabled, and an interrupt request is generated when the Debug Message Interrupt Flag is set.

35.7.5 Interrupt Flag Status and Clear register

Name: INTFLAG
Offset: 0x030
Reset: 0x00000000
Property: R/K

Table 35-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							BERR	DBG
Access							R/HS/K	R/HS/K
Reset							0	0

Bit 1 – BERR AHB Bus Error Detection

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the AHB Bus Error Detection Interrupt Flag.

Value	Description
0	The AHB Bus Error Detection Flag is cleared.
1	The AHB Bus Error Detection Flag is set.

Bit 0 – DBG Debug Message Reception

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Debug Message Interrupt Flag.

Value	Description
0	The Debug Message Flag is cleared.
1	The Debug Message Flag is set.

35.7.6 Interrupt Flag Software Set Register

Name: INTFLAGSET
Offset: 0x034
Reset: 0x00000000
Property: R/S

Table 35-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							BERR	DBG
Access							R/S	R/S
Reset							0	0

Bit 1 – BERR AHB Bus Error Detection Interrupt Flag Set

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the AHB Bus Error Detection Interrupt Flag.

Value	Description
0	The AHB Bus Error Detection Flag is cleared.
1	The AHB Bus Error Detection Flag is set.

Bit 0 – DBG Debug Message Interrupt Flag Set

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Debug Message Interrupt Flag.

Value	Description
0	The Debug Message Flag is cleared.
1	The Debug Message Flag is set.

35.7.7 SYNCBUSY Register

Name: SYNCBUSY
Offset: 0x05C
Reset: 0x00000000
Property: R

Table 35-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								SWRST
Access								R/HC/HS
Reset								0

Bit 0 – SWRST Software Reset Synchronization Busy

Value	Description
0	SWRST synchronization is not busy.
1	SWRST synchronization is busy.

35.7.8 Write Protection Control Register

Name: WPCTRL
Offset: 0x0E0
Reset: 0x00000000
Property: RW

Table 35-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPLCK	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Bit 1 – WPLCK Write Lock Bit

Bit 0 – WPEN Write Protection Enable

35.7.9 Core Release

Name: CREL_MCAN
Offset: 0x100
Reset: 0x33000000
Property: R

Table 35-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	REL[3:0]				STEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	1	0	0	1	1
Bit	23	22	21	20	19	18	17	16
	SUBSTEP[3:0]				YEAR[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MON[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DAY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – REL[3:0] Core Release

Bits 27:24 – STEP[3:0] Step of Core Release

Bits 23:20 – SUBSTEP[3:0] Sub-step of Core Release

Bits 19:16 – YEAR[3:0] Timestamp year

Bits 15:8 – MON[7:0] Timestamp month

Bits 7:0 – DAY[7:0] Timestamp day

35.7.10 Endian

Name: ENDN
Offset: 0x104
Reset: 0x87654321
Property: R

Table 35-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ETV[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	1	1	1
Bit	23	22	21	20	19	18	17	16
	ETV[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	0	0	1	0	1
Bit	15	14	13	12	11	10	9	8
	ETV[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	0	0	0	0	1	1
Bit	7	6	5	4	3	2	1	0
	ETV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	0	0	0	0	1

Bits 31:0 – ETV[31:0] Endianness Test Value
The endianness test value is 0x87654321

35.7.11 Customer Register

Name: CUST
Offset: 0x108
Reset: 0x00000000
Property: RW

Table 35-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CUST[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CUST[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CUST[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CUST[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CUST[31:0] Customer Register

35.7.12 Fast Bit Timing and Prescaler

Name: DBTP
Offset: 0x10C
Reset: 0x00000A33
Property: RW

This register is write-restricted and only writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

The CAN bit time may be programmed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 GCLK_CANx periods. time quantum (t_q) = (DBRP + 1) mtq. Therefore the length of the bit time is [DTSEG1 + DTSEG2 + 3] t_q where DTSEG1 and DTSEG2 are programmed values in the DBTP register.

Note:

With a GCLK_CANx of 8MHz, the reset value 0x00000A33 configures the CAN for a fast bit rate of 500 kBits/s.

The bit rate configured for the CAN FD data phase via the DBTP register must be higher or equal to the bit rate configured for the arbitration phase via the NBTP register.

Table 35-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
	TDC					DBRP[4:0]		
Access	RW			RW	RW	RW	RW	RW
Reset	0			0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						DTSEG1[4:0]		
Access				RW	RW	RW	RW	RW
Reset				0	1	0	1	0
Bit	7	6	5	4	3	2	1	0
						DTSEG2[3:0]		DSJW[3:0]
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	1	1	0	0	1	1

Bit 23 – TDC Transceiver Delay Compensation

Value	Description
0	Transceiver Delay Compensation disabled.
1	Transceiver Delay Compensation enabled.

Bits 20:16 – DBRP[4:0] Data Baud Rate Prescaler

Value	Description
0x00 – 0x1F	The value by which the GCLK_CANx is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 12:8 – DTSEG1[4:0] Data time segment before sample point

Value	Description
0x00 – 0x1F	Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG1 is the sum of PROPAGATION TIME SEGMENT (PROP_SEG) and PHASE BUFFER SEGMENT1 (PHASE_SEG1).

Bits 7:4 – DTSEG2[3:0] Data time segment after sample point

Value	Description
0x0 – 0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG2 is PHASE BUFFER SEGMENT2 (PHASE_SEG2).

Bits 3:0 – DSJW[3:0] Data (Re)Synchronization Jump Width

Value	Description
0x0 – 0xF	Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

35.7.13 Test

Name: TEST
Offset: 0x110
Reset: 0x00000000
Property: RW

Table 35-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	RX	TX[1:0]		LBCK				
Access	R	RW		RW				
Reset	0	0		0				

Bit 7 – RX Receive Pin

This bit reflects the actual value of pin CANx_RX. The read value can be interpreted as follows.

Value	Description
0	The CAN bus is dominant (CANx_RX = 0).
1	The CAN bus is recessive (CANx_RX = 1).

Bits 6:5 – TX[1:0] Control of Transmit Pin

This field defines the control of the transmit pin.

Value	Name	Description
0	CORE	TX controlled by CAN core
1	SAMPLE	TX monitoring sample point
2	DOMINANT	Dominant (0) level at pin CAN_TX
3	RECESSIVE	Recessive (1) level at pin CAN_TX

Bit 4 – LBCK Loop Back Mode

Value	Description
0	Loop Back Mode is disabled.
1	Loop Back Mode is enabled.

35.7.14 RAM Watchdog

Name: RWD
Offset: 0x114
Reset: 0x00000000
Property: RW

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the CAN's AHB Host Interface starts the Message RAM Watchdog Counter with the value configured by WDC bits (RWD <7:0>). The counter is reloaded with WDC bits (RWD <7:0>) when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt IR.WDI bit (IR<26>) is set.

Table 35-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	WDV[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	WDC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – WDV[7:0] Watchdog Value
Actual Message RAM Watchdog Counter Value.

Bits 7:0 – WDC[7:0] Watchdog Configuration
Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.

35.7.15 CC Control

Name: CCCR
Offset: 0x118
Reset: 0x00000001
Property: RW

Table 35-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
		TXP	EFBI	PXHD	WMM	UTSU	BRSE	FDOE
Access		RW	RW	RW	R/W	R/W	RW	RW
Reset		0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
Access	RW	RW	RW	RW	R	RW	RW	R/W
Reset	0	0	0	0	0	0	0	1

Bit 14 – TXP Transmit Pause

This bit field is write-restricted and only writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Transmit pause disabled.
1	Transmit pause enabled. The CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame.

Bit 13 – EFBI Edge Filtering during Bus Integration

Value	Description
0	Edge filtering is disabled.
1	Two consecutive dominant tq required to detect an edge for hard synchronization.

Bit 12 – PXHD Protocol Exception Handling Disable

Note: When protocol exception handling is disabled, the CAN will transmit an error frame when it detects a protocol exception condition.

Value	Description
0	Protocol exception handling enabled.
1	Protocol exception handling disabled.

Bit 11 – WMM Wide Message Marker

Bit 10 – UTSU Use Timestamping Unit

Value	Description
0	Use the internal Timestamp counter
1	Use the external Timestamp Unit

Bit 9 – BRSE Bit Rate Switch Enable

Note: When CAN FD operation is disabled (i.e., CCCR.FDOE bit (CCCR <8>) = 0), BRSE is not evaluated.

Value	Description
0	Bit rate switching for transmissions disabled.
1	Bit rate switching for transmissions enabled.

Bit 8 – FDOE FD Operation Enable

Value	Description
0	FD operation disabled.
1	FD operation enabled.

Bit 7 – TEST Test Mode Enable

This bit field is write-restricted.

Writing a 0 to this field is always allowed.

Writing a 1 to this field is allowed only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Normal operation. Register TEST holds reset values.
1	Test Mode, write access to register TEST enabled.

Bit 6 – DAR Disable Automatic Retransmission

This bit field is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Automatic retransmission of messages not transmitted successfully enabled.
1	Automatic retransmission disabled.

Bit 5 – MON Bus Monitoring Mode

This bit field is write-restricted.

Writing a 0 to this field is always allowed.

Writing a 1 to this field is allowed only both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Bus Monitoring Mode is disabled.
1	Bus Monitoring Mode is enabled.

Bit 4 – CSR Clock Stop Request

Value	Description
0	No clock stop is requested.
1	Clock stop requested. When clock stop is requested, first CCCR.INIT bit (CCCR <0>) and then the CCCR.CSA bit (CCCR <3>) will be set after all pending transfer requests have been completed and the CAN bus reached idle.

Bit 3 – CSA Clock Stop Acknowledge

Value	Description
0	No clock stop acknowledged.
1	CAN may be set in power down by stopping CLK_CANx_AHB and GCLK_CANx.

Bit 2 – ASM ASM Restricted Operation Mode

This bit field is write-restricted.

Writing a 0 to this field is always allowed.

Writing a 1 to this field is allowed only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Value	Description
0	Normal CAN operation.
1	Restricted Operation Mode active.

Bit 1 – CCE Configuration Change Enable

This bit field is write-restricted and only writable if bit field CCCR.INIT bit (CCCR <0>) is set.

Value	Description
0	The CPU has no write access to the protected configuration registers.
1	The CPU has write access to the protected configuration registers (while CCCR.INIT bit (CCCR <0>) =1).

Bit 0 – INIT Initialization

Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to the INIT bit can be read back. The programmer has to assure that the previous value written to the INIT bit has been accepted by reading the INIT bit before setting the INIT bit to a new value.

Value	Description
0	Normal Operation.
1	Initialization is started.

35.7.16 Nominal Bit Timing and Prescaler

Name: NBTP
Offset: 0x11C
Reset: 0x06000A03
Property: RW

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 GCLK_CANx periods. $t_q = (NBRP + 1) mtq$.

Therefore the length of the bit time is $[NTSEG1 + NTSEG2 + 3] t_q$, where NTSEG1 and NTSEG2 are programmed values in the NBTP register.

Note: With a CAN clock (GCLK_CANx) of 8MHz, the reset value 0x06000A03 configures the CAN for a bit rate of 500 kBits/s.

Table 35-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	NSJW[6:0]						NBRP[8]	
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	1	1	0
Bit	23	22	21	20	19	18	17	16
	NBRP[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NTSEG1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	1	0	1	0
Bit	7	6	5	4	3	2	1	0
		NTSEG2[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	1	1

Bits 31:25 – NSJW[6:0] Nominal (Re)Synchronization Jump Width

Value	Description
0x00 – 0x7F	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.

Bits 24:16 – NBRP[8:0] Nominal Baud Rate Prescaler

Value	Description
0x000 – 0x1FF	The value by which the GCLK_CANx is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 15:8 – NTSEG1[7:0] Nominal Time segment before sample point

Value	Description
0x00 – 0x7F	Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG1 is the sum of PROPAGATION TIME SEGMENT (PROP_SEG) and PHASE BUFFER SEGMENT1 (PHASE_SEG1).

Bits 6:0 – NTSEG2[6:0] Nominal Time segment after sample point

Value	Description
0x00 – 0x7F	Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG2 is PHASE BUFFER SEGMENT2 (PHASE_SEG2).

35.7.17 Timestamp Counter Configuration

Name: TSCC
Offset: 0x120
Reset: 0x00000000
Property: RW

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Table 35-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
					TCP[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							TSS[1:0]	
Access							RW	RW
Reset							0	0

Bits 19:16 – TCP[3:0] Timestamp Counter Prescaler

Value	Description
0x0 – 0xF	Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1...16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.

Bits 1:0 – TSS[1:0] Timestamp Select

This field defines the timestamp counter selection.

Value	Name	Description
0	ZERO	Timestamp counter value always 0x0000
1	INC	Timestamp counter value incremented by TCP
2	EXT	External timestamp counter value used

35.7.18 Timestamp Counter Value

Name: TSCV
Offset: 0x124
Reset: 0x00000000
Property: R

Notes:

1. A write access to TSCV clears the Timestamp Counter value.
2. A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by the write access to TSCV.

Table 35-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	TSC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TSC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – TSC[15:0] Timestamp Counter

The internal Timestamp Counter value is captured on start of frame (both Rx and Tx). When bitfield TSCC.TSS (TSCC<1:0>) = 0x1, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of bitfield TSCC.TCP (TSCC<19:16>). A wrap around sets interrupt flag IR.TSW (IR<16>).

35.7.19 Timeout Counter Configuration

Name: TOCC
Offset: 0x128
Reset: 0xFFFF0000
Property: RW

This register is write-restricted and writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

Table 35-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TOP[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	TOP[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						TOS[1:0]		ETOC
Access						RW	RW	RW
Reset						0	0	0

Bits 31:16 – TOP[15:0] Timeout Period

Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

Bits 2:1 – TOS[1:0] Timeout Select

When operating in Continuous mode, a write to TOCV register presets the counter to the value configured by bit TOCC.TOP (TOCC<31:16>) and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by bit TOCC.TOP (TOCC<31:16>). Down-counting is started when the first FIFO element is stored.

Value	Name	Description
0	CONT	Continuout operation
1	TXEF	Timeout controlled by TX Event FIFO
2	RXF0	Timeout controlled by Rx FIFO 0
3	RXF1	Timeout controlled by Rx FIFO 1

Bit 0 – ETOC Enable Timeout Counter

Value	Description
0	Timeout Counter disabled.

Value	Description
1	Timeout Counter enabled.

35.7.20 Timeout Counter Value

Name: TOCV
Offset: 0x12C
Reset: 0x0000FFFF
Property: RW

Note: A write access to TOCV reloads the Timeout Counter with the value of bit TOCC.TOP (TOCC<31:16>).

Table 35-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	TOC[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	TOC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1

Bits 15:0 – TOC[15:0] Timeout Counter

The Timeout Counter is decremented in multiples of CAN bit times [1...16] depending on the configuration of the TSCC.TCP bit (TSCC<19:16>). When decremented to zero, interrupt flag (IR.TOO bit (IR<18>)) is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS bit (TOCC<2:1>).

35.7.21 Error Counter

Name: ECR
Offset: 0x140
Reset: 0x00000000
Property: R

Note: When CCCR.ASM bit (CCCR<2>) is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

Table 35-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	CEL[7:0]							
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	RP	REC[6:0]						
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	TEC[7:0]							
Reset	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – CEL[7:0] CAN Error Logging

The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; and the next increment of TEC or REC sets interrupt flag IR.ELO bit (IR<22>).

Bit 15 – RP Receive Error Passive

Bits 14:8 – REC[6:0] Receive Error Counter

Actual state of the Receive Error Counter, values between 0 and 127.

Bits 7:0 – TEC[7:0] Transmit Error Counter

Actual state of the Transmit Error Counter, values between 0 and 255.

35.7.22 Protocol Status

Name: PSR
Offset: 0x144
Reset: 0x00000707
Property: R

Notes:

1. When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in PSR.DLEC bit field (PSR<10:8>) instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.
2. The 'bus off' recovery sequence (see CAN Specification Rev. 2.0 or ISO 11898-1) cannot be shortened by setting or resetting CCCR.INIT bit (CCCR <0>). If the device goes 'bus off', it will set CCCR.INIT bit (CCCR <0>) of its own accord, stopping all bus activities. Once CCCR.INIT bit (CCCR <0>) has been cleared by the CPU, the device will wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the 'bus off' recovery sequence, the Error Management Counters will be reset. During the wait time after the resetting of CCCR.INIT bit (CCCR <0>), each time a sequence of 11 recessive bits is monitored, a Bit0 Error code is written to PSR.LEC bit field (PSR <2:0>). This enables the CPU to readily check the status of CAN bus (whether bus is stuck at dominant level or continuously disturbed). ECR.REC bit-field (ECR<14:8>) is used to count these sequences.

Table 35-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
		TDCV[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
		PXE	RFDF	RBRS	RESI	DLEC[2:0]		
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	1	1	1
Bit	7	6	5	4	3	2	1	0
	BO	EW	EP	ACT[1:0]		LEC[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	1	1	1

Bits 22:16 – TDCV[6:0] Transmitter Delay Compensation Value

Value	Description
0x00 – 0x7F	Position of the secondary sample point, defined by the sum of the measured delay from CANx_TX to CANx_RX and TDCR.TDCO bit-field (TDCR <14:8>). The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.

Bit 14 – PXE Protocol Exception Event

This field is cleared on read access. A recessive “reserved bit” following a recessive FDF bit is the example of Protocol Exception.

Value	Description
0	No protocol exception event occurred since last read access.
1	Protocol exception event occurred.

Bit 13 – RFDF Received a CAN FD Message

This field is cleared on read access.

Value	Description
0	Since this bit was reset by the CPU, no CAN FD message has been received.
1	Message in CAN FD format with FDF flag set has been received. This bit is set independent of acceptance filtering.

Bit 12 – RBRS BRS flag of last received CAN FD Message

This field is cleared on read access.

Value	Description
0	Last received CAN FD message did not have its BRS flag set.
1	Last received CAN FD message had its BRS flag set. This bit is set together with RFDF, independent of acceptance filtering.

Bit 11 – RESI ESI flag of last received CAN FD Message

This field is cleared on read access.

Value	Description
0	Last received CAN FD message did not have its ESI flag set.
1	Last received CAN FD message had its ESI flag set.

Bits 10:8 – DLEC[2:0] Data Phase Last Error Code

Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

Value	Name	Description
0	NONE	No Error
1	STUFF	Stuff Error
2	FORM	Form Error
3	ACK	Ack Error
4	BIT1	Bit1 Error
5	BIT0	Bit0 Error
6	CRC	CRC Error
7	NC	No Change

Bit 7 – BO Bus_Off Status

Value	Description
0	The CAN is not 'bus off' state.
1	The CAN is in 'bus off' state.

Bit 6 – EW Warning Status

Value	Description
0	Both error counters are below the Error Warning limit of 96.
1	At least one of the error counter has reached the Error Warning limit of 96.

Bit 5 – EP Error Passive

Value	Description
0	The CAN is in the 'error active' state. It normally takes part in bus communication and sends an active error flag when an error has been detected.
1	The CAN is in the 'error passive' state.

Bits 4:3 – ACT[1:0] Activity

Monitors the module's CAN communication state.

Value	Name	Description
0	SYNC	Node is synchronizing on CAN communication
1	IDLE	Node is neither receiver nor transmitter
2	RX	Node is operating as receiver
3	TX	Node is operating as transmitter

Bits 2:0 – LEC[2:0] Last Error Code

The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. This field is set on read access.

Value	Name	Description
0	NONE	No Error
1	STUFF	Stuff Error
2	FORM	Form Error
3	ACK	Ack Error
4	BIT1	Bit1 Error
5	BIT0	Bit0 Error
6	CRC	CRC Error
7	NC	No Change

35.7.23 Extended ID Filter Configuration

Name: TDCR
Offset: 0x148
Reset: 0x00000000
Property: RW

Table 35-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
		TDCO[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
		TDCF[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 14:8 – TDCO[6:0] Transmitter Delay Compensation Offset

Value	Description
0x00 – 0x7F	Offset value defining the distance between the measured delay from CANx_TX to CANx_RX and the secondary sample point. Valid values are 0 to 127 mtq.

Bits 6:0 – TDCF[6:0] Transmitter Delay Compensation Filter Length

Value	Description
0x00 – 0x7F	Defines the minimum value for the SSP position, dominant edges on CANx_RX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.

35.7.24 Interrupt

Name: IR
Offset: 0x150
Reset: 0x00000000
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

The flags are set when one of the listed conditions is detected (edge-sensitive). A flag is cleared by writing a 1 to the corresponding bit field. Writing a 0 has no effect. A hard reset will clear the register.

Table 35-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			ARA	PED	PEA	WDI	BO	EW
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARA Access to Reserved Address

Value	Description
0	No access to reserved address occurred.
1	Access to reserved address occurred.

Bit 28 – PED Protocol Error in Data Phase

Value	Description
0	No protocol error in data phase detected.
1	Protocol error in data phase detected (i.e., PSR.DLEC bits (PSR<10:8>) != 0 and PSR.DLEC bits (PSR<10:8>) != 0).

Bit 27 – PEA Protocol Error in Arbitration Phase

Value	Description
0	No protocol error in arbitration phase detected.
1	Protocol error in arbitration phase detected (i.e., PSR.LEC bits (PSR<2:0>) != 0 and PSR.LEC bits (PSR<2:0>) != 0).

Bit 26 – WDI Watchdog Interrupt

Value	Description
0	No Message RAM Watchdog event occurred.
1	Message RAM Watchdog event due to missing READY.

Bit 25 – BO Bus_Off Status

Value	Description
0	'bus off' status unchanged.
1	'bus off' status changed.

Bit 24 – EW Warning Status

Value	Description
0	Error Warning status unchanged.
1	Error Warning status changed.

Bit 23 – EP Error Passive

Value	Description
0	Error Passive status unchanged.
1	Error Passive status changed.

Bit 22 – ELO Error Logging Overflow

Value	Description
0	CAN Error Logging Counter did not overflow.
1	Overflow of CAN Error Logging Counter occurred.

Bit 21 – BEU Bit Error Uncorrected

Bit 20 – BEC Bit Error Corrected

Bit 19 – DRX Message stored to Dedicated Rx Buffer

The flag is set whenever a received message has been stored into a dedicated Rx Buffer.

Value	Description
0	No Rx Buffer updated.
1	At least one received message stored into a Rx Buffer.

Bit 18 – TOO Timeout Occurred

Value	Description
0	No timeout.
1	Timeout reached.

Bit 17 – MRAF Message RAM Access Failure

The flag is set, when the Rx Handler:

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.

- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated. The New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU must clear CCCR.ASM bit (CCCR <2>).

Value	Description
0	No Message RAM access failure occurred.
1	Message RAM access failure occurred.

Bit 16 – TSW Timestamp Wraparound

Value	Description
0	No timestamp counter wrap-around.
1	Timestamp counter wrapped around.

Bit 15 – TEFL Tx Event FIFO Element Lost

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 14 – TEFF Tx Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bit 13 – TEFW Tx Event FIFO Watermark Reached

Value	Description
0	Tx Event FIFO fill level below watermark.
1	Tx Event FIFO fill level reached watermark.

Bit 12 – TEFN Tx Event FIFO New Entry

Value	Description
0	Tx Event FIFO unchanged.
1	Tx Handler wrote Tx Event FIFO element.

Bit 11 – TFE Tx FIFO Empty

Value	Description
0	Tx FIFO not-empty.
1	Tx FIFO empty.

Bit 10 – TCF Transmission Cancellation Finished

Value	Description
0	Transmission cancellation not finished.
1	Transmission cancellation finished.

Bit 9 – TC Timestamp Completed

Value	Description
0	No transmission completed.

Value	Description
1	Transmission completed.

Bit 8 – HPM High Priority Message

Value	Description
0	No high priority message received.
1	High priority message received.

Bit 7 – RF1L Rx FIFO 1 Message Lost

Value	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost. also set after write attempt to Rx FIFO 1 of size zero.

Bit 6 – RF1F Rx FIFO 1 FIFO Full

Value	Description
0	Rx FIFO 1 not full.
1	Rx FIFO 1 full.

Bit 5 – RF1W Rx FIFO 1 Watermark Reached

Value	Description
0	Rx FIFO 1 fill level below watermark.
1	Rx FIFO 1 fill level reached watermark.

Bit 4 – RF1N Rx FIFO 1 New Message

Value	Description
0	No new message written to Rx FIFO 1.
1	New message written to Rx FIFO 1.

Bit 3 – RF0L Rx FIFO 0 Message Lost

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost. also set after write attempt to Rx FIFO 0 of size zero.

Bit 2 – RF0F Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bit 1 – RF0W Rx FIFO 0 Watermark Reached

Value	Description
0	Rx FIFO 0 fill level below watermark.
1	Rx FIFO 0 fill level reached watermark.

Bit 0 – RF0N Rx FIFO 0 New Message

Value	Description
0	No new message written to Rx FIFO 0.
1	New message written to Rx FIFO 0.

35.7.25 Interrupt Enable

Name: IE
Offset: 0x154
Reset: 0x00000000
Property: RW

Table 35-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			ARAE	PEDE	PEAE	WDIE	BOE	EWE
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAE Access to Reserved Address Enable

Bit 28 – PEDE Protocol Error in Data Phase Enable

Bit 27 – PEAE Protocol Error in Arbitration Phase Enable

Bit 26 – WDIE Watchdog Interrupt Interrupt Enable

Bit 25 – BOE Bus_Off Status Interrupt Enable

Bit 24 – EWE Warning Status Interrupt Enable

Bit 23 – EPE Error Passive Interrupt Enable

Bit 22 – ELOE Error Logging Overflow Interrupt Enable

Bit 21 – BEUE Bit Error Uncorrected Interrupt Enable

Bit 20 – BECE Bit Error Corrected Interrupt Enable

Bit 19 – DRXE Message stored to Dedicated Rx Buffer Interrupt Enable

Bit 18 – TOOE Timeout Occurred Interrupt Enable

Bit 17 – MRAFE Message RAM Access Failure Interrupt Enable

Bit 16 – TSWE Timestamp Wraparound Interrupt Enable

Bit 15 – TEFLE Tx Event FIFO Element Lost Interrupt Enable

Bit 14 – TEFFE Tx Event FIFO Full Interrupt Enable

Bit 13 – TEFWE Tx Event FIFO Watermark Reached Interrupt Enable

Bit 12 – TEFNE Tx Event FIFO New Entry Interrupt Enable

Bit 11 – TFEE Tx FIFO Empty Interrupt Enable

Bit 10 – TCFE Transmission Cancellation Finished Interrupt Enable

Bit 9 – TCE Timestamp Completed Interrupt Enable

Bit 8 – HPME High Priority Message Interrupt Enable

Bit 7 – RF1LE Rx FIFO 1 Message Lost Interrupt Enable

Bit 6 – RF1FE Rx FIFO 1 FIFO Full Interrupt Enable

Bit 5 – RF1WE Rx FIFO 1 Watermark Reached Interrupt Enable

Bit 4 – RF1NE Rx FIFO 1 New Message Interrupt Enable

Bit 3 – RF0LE Rx FIFO 0 Message Lost Interrupt Enable

Bit 2 – RF0FE Rx FIFO 0 Full Interrupt Enable

Bit 1 – RF0WE Rx FIFO 0 Watermark Reached Interrupt Enable

Bit 0 – RF0NE Rx FIFO 0 New Message Interrupt Enable

35.7.26 Interrupt Line Select

Name: ILS
Offset: 0x158
Reset: 0x00000000
Property: RW

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from IR to one of the two module interrupt lines.

Table 35-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			ARAL	PEDL	PEAL	WDIL	BOL	EWL
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RF1LL	RF1FL	RF1WL	RF1NL	RFOLL	RF0FL	RF0WL	RF0NL
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 29 – ARAL Access to Reserved Address Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 28 – PEDL Protocol Error in Data Phase Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 27 – PEAL Protocol Error in Arbitration Phase Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 26 – WDIL Watchdog Interrupt Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 25 – BOL Bus_Off Status Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 24 – EWL Warning Status Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 23 – EPL Error Passive Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 22 – ELOL Error Logging Overflow Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 21 – BEUL Bit Error Uncorrected Interrupt Line

Bit 20 – BECL Bit Error Corrected Interrupt Line

Bit 19 – DRXL Message stored to Dedicated Rx Buffer Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 18 – TOOL Timeout Occurred Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 17 – MRAFL Message RAM Access Failure Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 16 – TSWL Timestamp Wraparound Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 15 – TEFL Tx Event FIFO Element Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 14 – TEFFL Tx Event FIFO Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 13 – TEFWL Tx Event FIFO Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 12 – TEFNL Tx Event FIFO New Entry Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 11 – TFEL Tx FIFO Empty Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 10 – TCFL Transmission Cancellation Finished Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 9 – TCL Timestamp Completed Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 8 – HPML High Priority Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 7 – RF1LL Rx FIFO 1 Message Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 6 – RF1FL Rx FIFO 1 FIFO Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 5 – RF1WL Rx FIFO 1 Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 4 – RF1NL Rx FIFO 1 New Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 3 – RF0LL Rx FIFO 0 Message Lost Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 2 – RF0FL Rx FIFO 0 Full Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 1 – RF0WL Rx FIFO 0 Watermark Reached Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

Bit 0 – RF0NL Rx FIFO 0 New Message Interrupt Line

Value	Description
0	Interrupt assigned to CAN interrupt line 0.
1	Interrupt assigned to CAN interrupt line 1.

35.7.27 Interrupt Line Enable

Name: ILE
Offset: 0x15C
Reset: 0x00000000
Property: RW

Table 35-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							EINT1	EINT0
Access							RW	RW
Reset							0	0

Bit 1 – EINT1 Enable Interrupt Line 1

Value	Description
0	CAN interrupt line 1 disabled.
1	CAN interrupt line 1 enabled.

Bit 0 – EINT0 Enable Interrupt Line 0

Value	Description
0	CAN interrupt line 0 disabled.
1	CAN interrupt line 0 enabled.

35.7.28 Global Filter Configuration

Name: GFC
Offset: 0x180
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 35-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:4 – ANFS[1:0] Accept Non-matching Frames Standard

Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RXF0	Accept in Rx FIFO 0
1	RXF1	Accept in Rx FIFO 1
2	REJECT	Reject

Bits 3:2 – ANFE[1:0] Accept Non-matching Frames Extended

Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

Value	Name	Description
0	RXF0	Accept in Rx FIFO 0
1	RXF1	Accept in Rx FIFO 1
2	REJECT	Reject

Bit 1 – RRFS Reject Remote Frames Standard

Value	Description
0	Filter remote frames with 11-bit standard IDs.
1	Reject all remote frames with 11-bit standard IDs.

Bit 0 – RRFE Reject Remote Frames Extended

Value	Description
0	Filter remote frames with 29-bit extended IDs.
1	Reject all remote frames with 29-bit extended IDs.

35.7.29 Standard ID Filter Configuration

Name: SIDFC
Offset: 0x184
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 35-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	LSS[7:0]							
Reset	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	FLSSA[13:6]							
Reset	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	FLSSA[5:0]							
Reset	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bits 23:16 – LSS[7:0] List Size Standard

Value	Description
0	No standard Message ID filter.
1 – 128	Number of standard Message ID filter elements.
> 128	Values greater than 128 are interpreted as 128.

Bits 15:2 – FLSSA[13:0] Filter List Standard Start Address

Start address of standard Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.7.30 Extended ID Filter Configuration

Name: XIDFC
Offset: 0x188
Reset: 0x00000000
Property: RW

Table 35-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
		LSE[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FLESA[13:6]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FLESA[5:0]							
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bits 22:16 – LSE[6:0] List Size Extended

Value	Description
0	No extended Message ID filter.
1 – 64	Number of Extended Message ID filter elements.
> 64	Values greater than 64 are interpreted as 64.

Bits 15:2 – FLESA[13:0] Filter List Extended Start Address

Start address of extended Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.7.31 Extended ID AND Mask

Name: XIDAM
Offset: 0x190
Reset: 0x1FFFFFFF
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 35-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				EIDM[28:24]				
Access				RW	RW	RW	RW	RW
Reset				1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	EIDM[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	EIDM[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	EIDM[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1

Bits 28:0 – EIDM[28:0] Extended ID Mask

For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

35.7.32 High Priority Message Status

Name: HPMS
Offset: 0x194
Reset: 0x00000000
Property: R

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Table 35-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	FLST	FIDX[6:0]						
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MSI[1:0]		BIDX[5:0]					
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FLST Filter List

Indicates the filter list of the matching filter element.

Value	Description
0	Standard Filter List.
1	Extended Filter List.

Bits 14:8 – FIDX[6:0] Filter Index

Index of matching filter element. Range is 0 to SIDFC.LSS bits (SIDFC <23:16>) bits (XIDFC <22:16>) - 1 (standard) or XIDFC.LSE bits (XIDFC <22:16>)- 1 (extended).

Bits 7:6 – MSI[1:0] Message Storage Indicator

This field defines the message storage information to a FIFO.

Value	Name	Description
0	NONE	No FIFO selected
1	LOST	FIFO message lost
2	FIFO0	Message stored in FIFO 0

Value	Name	Description
3	FIFO1	Message stored in FIFO 1

Bits 5:0 – BIDX[5:0] Buffer Index

Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.

35.7.33 New Data 1

Name: NDAT1
Offset: 0x198
Reset: 0x00000000
Property: RW

Table 35-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – ND31 New Data 31

Bit 30 – ND30 New Data 30

Bit 29 – ND29 New Data 29

Bit 28 – ND28 New Data 28

Bit 27 – ND27 New Data 27

Bit 26 – ND26 New Data 26

Bit 25 – ND25 New Data 25

Bit 24 – ND24 New Data 24

Bit 23 – ND23 New Data 23

Bit 22 – ND22 New Data 22

Bit 21 – ND21 New Data 21

Bit 20 – ND20 New Data 20

Bit 19 – ND19 New Data 19

Bit 18 – ND18 New Data 18

Bit 17 – ND17 New Data 17

Bit 16 – ND16 New Data 16

Bit 15 – ND15 New Data 15

Bit 14 – ND14 New Data 14

Bit 13 – ND13 New Data 13

Bit 12 – ND12 New Data 12

Bit 11 – ND11 New Data 11

Bit 10 – ND10 New Data 10

Bit 9 – ND9 New Data 9

Bit 8 – ND8 New Data 8

Bit 7 – ND7 New Data 7

Bit 6 – ND6 New Data 6

Bit 5 – ND5 New Data 5

Bit 4 – ND4 New Data 4

Bit 3 – ND3 New Data 3

Bit 2 – ND2 New Data 2

Bit 1 – ND1 New Data 1

Bit 0 – ND0 New Data 0

35.7.34 New Data 2

Name: NDAT2
Offset: 0x19C
Reset: 0x00000000
Property: RW

Table 35-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – ND63 New Data 63

Bit 30 – ND62 New Data 62

Bit 29 – ND61 New Data 61

Bit 28 – ND60 New Data 60

Bit 27 – ND59 New Data 59

Bit 26 – ND58 New Data 58

Bit 25 – ND57 New Data 57

Bit 24 – ND56 New Data 56

Bit 23 – ND55 New Data 55

Bit 22 – ND54 New Data 54

Bit 21 – ND53 New Data 53

Bit 20 – ND52 New Data 52

Bit 19 – ND51 New Data 51

Bit 18 – ND50 New Data 50

Bit 17 – ND49 New Data 49

Bit 16 – ND48 New Data 48

Bit 15 – ND47 New Data 47

Bit 14 – ND46 New Data 46

Bit 13 – ND45 New Data 45

Bit 12 – ND44 New Data 44

Bit 11 – ND43 New Data 43

Bit 10 – ND42 New Data 42

Bit 9 – ND41 New Data 41

Bit 8 – ND40 New Data 40

Bit 7 – ND39 New Data 39

Bit 6 – ND38 New Data 38

Bit 5 – ND37 New Data 37

Bit 4 – ND36 New Data 36

Bit 3 – ND35 New Data 35

Bit 2 – ND34 New Data 34

Bit 1 – ND33 New Data 33

Bit 0 – ND32 New Data 32

35.7.35 Rx FIFO 0 Configuration

Name: RXF0C
Offset: 0x1A0
Reset: 0x00000000
Property: RW

Table 35-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	F00M	F0WM[6:0]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
		F0S[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	F0SA[13:6]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	F0SA[5:0]							
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bit 31 – F00M FIFO 0 Operation Mode

FIFO 0 can be operated in blocking or in overwrite mode.

Value	Description
0	FIFO 0 blocking mode.
1	FIFO 0 overwrite mode.

Bits 30:24 – F0WM[6:0] Rx FIFO 0 Watermark

Value	Description
0	Watermark interrupt disabled.
1 – 64	Level for Rx FIFO 0 watermark interrupt (IR.RF0W bit (IR <1>)).
>64	Watermark interrupt disabled.

Bits 22:16 – F0S[6:0] Rx FIFO 0 Size

The Rx FIFO 0 elements are indexed from 0 to F0S - 1.

Value	Description
0	No Rx FIFO 0
1 – 64	Number of Rx FIFO 0 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:2 – F0SA[13:0] Rx FIFO 0 Start Address

Start address of Rx FIFO 0 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.7.36 Rx FIFO 0 Status

Name: RXF0S
Offset: 0x1A4
Reset: 0x00000000
Property: R

Table 35-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							RF0L	F0F
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
			F0PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F0GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F0FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit 25 – RF0L Rx FIFO 0 Message Lost

This bit is a copy of interrupt flag IR.RF0L bit (IR <3>). When IR.RF0L bit (IR <3>) is reset, this bit is also reset.

Overwriting the oldest message when RXF0C.F0OM bit (RXF0C <31>) = '1' will not set this flag.

Value	Description
0	No Rx FIFO 0 message lost.
1	Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero.

Bit 24 – F0F Rx FIFO 0 Full

Value	Description
0	Rx FIFO 0 not full.
1	Rx FIFO 0 full.

Bits 21:16 – F0PI[5:0] Rx FIFO 0 Put Index

Rx FIFO 0 write index pointer, range 0 to 63.

Bits 13:8 – F0GI[5:0] Rx FIFO 0 Get Index

Rx FIFO 0 read index pointer, range 0 to 63.

Bits 6:0 – F0FL[6:0] Rx FIFO 0 Fill Level
Number of elements stored in Rx FIFO 0, range 0 to 64.

35.7.37 Rx FIFO 0 Acknowledge

Name: RXF0A
Offset: 0x1A8
Reset: 0x00000000
Property: RW

Table 35-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			F0AI[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:0 – F0AI[5:0] Rx FIFO 0 Acknowledge Index

After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI bits (RXF0S <13:8>) to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL bit (RXF0S <25>).

35.7.38 Rx Buffer Configuration

Name: RXBC
Offset: 0x1AC
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 35-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	RBSA[13:6]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RBSA[5:0]							
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bits 15:2 – RBSA[13:0] Rx Buffer Start Address

Configures the start address of the Rx Buffers section in the Message RAM. Also used to reference debug message A,B,C. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.7.39 Rx FIFO 1 Configuration

Name: RXF1C
Offset: 0x1B0
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 35-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	F1OM	F1WM[6:0]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
		F1S[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	F1SA[13:6]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	F1SA[5:0]							
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bit 31 – F1OM FIFO 1 Operation Mode

FIFO 1 can be operated in blocking or in overwrite mode.

Value	Description
0	FIFO 1 blocking mode.
1	FIFO 1 overwrite mode.

Bits 30:24 – F1WM[6:0] Rx FIFO 1 Watermark

Value	Description
0	Watermark interrupt disabled.
1 – 64	Level for Rx FIFO 1 watermark interrupt (IR.RF1W bit (IR <5>)).
>64	Watermark interrupt disabled.

Bits 22:16 – F1S[6:0] Rx FIFO 1 Size

The Rx FIFO 1 elements are indexed from 0 to F1S - 1.

Value	Description
0	No Rx FIFO 1

Value	Description
1 – 64	Number of Rx FIFO 1 elements.
>64	Values greater than 64 are interpreted as 64.

Bits 15:2 – F1SA[13:0] Rx FIFO 1 Start Address

Start address of Rx FIFO 1 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.7.40 Rx FIFO 1 Status

Name: RXF1S
Offset: 0x1B4
Reset: 0x00000000
Property: R

Table 35-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DMS[1:0]						RF1L	F1F
Access	R	R					R	R
Reset	0	0					0	0

Bit	23	22	21	20	19	18	17	16
			F1PI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
			F1GI[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		F1FL[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bits 31:30 – DMS[1:0] Debug Message Status

This field defines the debug message status.

Value	Name	Description
0	IDLE	Idle state
1	DBGA	Debug message A received
2	DBGB	Debug message A/B received
3	DBGC	Debug message A/B/C received, DMA request set

Bit 25 – RF1L Rx FIFO 1 Message Lost

This bit is a copy of interrupt flag IR.RF1L bit (IR <7>). When IR.RF1L bit (IR <7>) is reset, this bit is also reset.

Overwriting the oldest message when RXF1C.F1OM bit (RXF1C <31>) = '1' will not set this flag.

Value	Description
0	No Rx FIFO 1 message lost.
1	Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero.

Bit 24 – F1F Rx FIFO 1 Full

Value	Description
0	Rx FIFO 1 not full.

Value	Description
1	Rx FIFO 1 full.

Bits 21:16 – F1PI[5:0] Rx FIFO 1 Put Index
Rx FIFO 1 write index pointer, range 0 to 63.

Bits 13:8 – F1GI[5:0] Rx FIFO 1 Get Index
Rx FIFO 1 read index pointer, range 0 to 63.

Bits 6:0 – F1FL[6:0] Rx FIFO 1 Fill Level
Number of elements stored in Rx FIFO 1, range 0 to 64.

35.7.41 Rx FIFO 1 Acknowledge

Name: RXF1A
Offset: 0x1B8
Reset: 0x00000000
Property: RW

Table 35-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			F1AI[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:0 – F1AI[5:0] Rx FIFO 1 Acknowledge Index

After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI bits (RXF1S <13:8>) to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL bits (RXF1S <6:0>).

35.7.42 Rx Buffer / FIFO Element Size Configuration

Name: RXESC
Offset: 0x1BC
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Configures the number of data bytes belonging to an Rx Buffer/Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

Table 35-60. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							RBDS[2:0]	
Access						RW	RW	RW
Reset						0	0	0
Bit	7	6	5	4	3	2	1	0
		F1DS[2:0]				F0DS[2:0]		
Access		RW	RW	RW		RW	RW	RW
Reset		0	0	0		0	0	0

Bits 10:8 – RBDS[2:0] Rx Buffer Data Field Size

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer, only the number of bytes as configured by RXESC are stored to the Rx Buffer element. The rest of the frame's data field is ignored.

Value	Name	Description
0	DATA8	8 byte data field
1	DATA12	12 byte data field
2	DATA16	16 byte data field
3	DATA20	20 byte data field
4	DATA24	24 byte data field
5	DATA32	32 byte data field
6	DATA48	48 byte data field
7	DATA64	64 byte data field

Bits 6:4 – F1DS[2:0] Rx FIFO 1 Data Field Size

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 1, only the number of bytes as configured by RXESC are stored to the Rx FIFO 1 element. The rest of the frame's data field is ignored.

Value	Name	Description
0	DATA8	8 byte data field
1	DATA12	12 byte data field
2	DATA16	16 byte data field
3	DATA20	20 byte data field
4	DATA24	24 byte data field
5	DATA32	32 byte data field
6	DATA48	48 byte data field
7	DATA64	64 byte data field

Bits 2:0 – F0DS[2:0] Rx FIFO 0 Data Field Size

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 0, only the number of bytes as configured by RXESC are stored to the Rx FIFO 0 element. The rest of the frame's data field is ignored.

Value	Name	Description
0	DATA8	8 byte data field
1	DATA12	12 byte data field
2	DATA16	16 byte data field
3	DATA20	20 byte data field
4	DATA24	24 byte data field
5	DATA32	32 byte data field
6	DATA48	48 byte data field
7	DATA64	64 byte data field

35.7.43 Tx Buffer Configuration

Name: TXBC
Offset: 0x1C0
Reset: 0x00000000
Property: RW

Table 35-61. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		TFQM				TFQS[5:0]		
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
						NDTB[5:0]		
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						TBSA[13:6]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						TBSA[5:0]		
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bit 30 – TFQM Tx FIFO/Queue Mode

Bits 29:24 – TFQS[5:0] Transmit FIFO/Queue Size

Bits 21:16 – NDTB[5:0] Number of Dedicated Transmit Buffers

Bits 15:2 – TBSA[13:0] Tx Buffers Start Address

35.7.44 Tx FIFO / Queue Status

Name: TXFQS
Offset: 0x1C4
Reset: 0x00000000
Property: R

Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indexes indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. For example, for a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

Table 35-62. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			TFQF			TFQPI[4:0]		
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						TFGI[4:0]		
Access				R	R	R	R	R
Reset				0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						TFFL[5:0]		
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 21 – TFQF Tx FIFO/Queue Full

Value	Description
0	Tx FIFO/Queue not full.
1	Tx FIFO/Queue full.

Bits 20:16 – TFQPI[4:0] Tx FIFO/Queue Put Index Tx FIFO/Queue write index pointer, range 0 to 31.

Bits 12:8 – TFGI[4:0] Tx FIFO Get Index Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM bit (TXBC <30>) = '1').

Bits 5:0 – TFFL[5:0] Tx FIFO Free Level Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM bit (TXBC <30>) = '1').

35.7.45 Tx Buffer Element Size Configuration

Name: TXESC
Offset: 0x1C8
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes >8 bytes are intended for CAN FD operation only.

Table 35-63. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						TBDS[2:0]		
Access						RW	RW	RW
Reset						0	0	0

Bits 2:0 – TBDS[2:0] Tx Buffer Data Field Size

In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS (TXESC <2:0>), the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

Value	Name	Description
0	DATA8	8 byte data field
1	DATA12	12 byte data field
2	DATA16	16 byte data field
3	DATA20	20 byte data field
4	DATA24	24 byte data field
5	DATA32	32 byte data field
6	DATA48	48 byte data field
7	DATA64	64 byte data field

35.7.46 Tx Buffer Request Pending

Name: TXBRP
Offset: 0x1CC
Reset: 0x00000000
Property: R

Note: The TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is canceled immediately, the corresponding TXBRP bit is reset.

Table 35-64. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – TRP31 Transmission Request Pending 31

Bit 30 – TRP30 Transmission Request Pending 30

Bit 29 – TRP29 Transmission Request Pending 29

Bit 28 – TRP28 Transmission Request Pending 28

Bit 27 – TRP27 Transmission Request Pending 27

Bit 26 – TRP26 Transmission Request Pending 26

Bit 25 – TRP25 Transmission Request Pending 25

Bit 24 – TRP24 Transmission Request Pending 24

Bit 23 – TRP23 Transmission Request Pending 23

Bit 22 – TRP22 Transmission Request Pending 22

Bit 21 – TRP21 Transmission Request Pending 21

Bit 20 – TRP20 Transmission Request Pending 20

Bit 19 – TRP19 Transmission Request Pending 19

Bit 18 – TRP18 Transmission Request Pending 18

Bit 17 – TRP17 Transmission Request Pending 17

Bit 16 – TRP16 Transmission Request Pending 16

Bit 15 – TRP15 Transmission Request Pending 15

Bit 14 – TRP14 Transmission Request Pending 14

Bit 13 – TRP13 Transmission Request Pending 13

Bit 12 – TRP12 Transmission Request Pending 12

Bit 11 – TRP11 Transmission Request Pending 11

Bit 10 – TRP10 Transmission Request Pending 10

Bit 9 – TRP9 Transmission Request Pending 9

Bit 8 – TRP8 Transmission Request Pending 8

Bit 7 – TRP7 Transmission Request Pending 7

Bit 6 – TRP6 Transmission Request Pending 6

Bit 5 – TRP5 Transmission Request Pending 5

Bit 4 – TRP4 Transmission Request Pending 4

Bit 3 – TRP3 Transmission Request Pending 3

Bit 2 – TRP2 Transmission Request Pending 2

Bit 1 – TRP1 Transmission Request Pending 1

Bit 0 – TRP0 Transmission Request Pending 0

35.7.47 Tx Buffer Add Request

Name: TXBAR
Offset: 0x1D0
Reset: 0x00000000
Property: RW

Note: If an add request is applied for a Tx Buffer with pending transmission request (the corresponding TXBRP bit is already set), this add request is ignored.

Table 35-65. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – AR31 Add Request 31

Bit 30 – AR30 Add Request 30

Bit 29 – AR29 Add Request 29

Bit 28 – AR28 Add Request 28

Bit 27 – AR27 Add Request 27

Bit 26 – AR26 Add Request 26

Bit 25 – AR25 Add Request 25

Bit 24 – AR24 Add Request 24

Bit 23 – AR23 Add Request 23

Bit 22 – AR22 Add Request 22

Bit 21 – AR21 Add Request 21

Bit 20 – AR20 Add Request 20

Bit 19 – AR19 Add Request 19

Bit 18 – AR18 Add Request 18

Bit 17 – AR17 Add Request 17

Bit 16 – AR16 Add Request 16

Bit 15 – AR15 Add Request 15

Bit 14 – AR14 Add Request 14

Bit 13 – AR13 Add Request 13

Bit 12 – AR12 Add Request 12

Bit 11 – AR11 Add Request 11

Bit 10 – AR10 Add Request 10

Bit 9 – AR9 Add Request 9

Bit 8 – AR8 Add Request 8

Bit 7 – AR7 Add Request 7

Bit 6 – AR6 Add Request 6

Bit 5 – AR5 Add Request 5

Bit 4 – AR4 Add Request 4

Bit 3 – AR3 Add Request 3

Bit 2 – AR2 Add Request 2

Bit 1 – AR1 Add Request 1

Bit 0 – AR0 Add Request 0

35.7.48 Tx Buffer Cancellation Request

Name: TXBCR
Offset: 0x1D4
Reset: 0x00000000
Property: RW

Table 35-66. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – CR31 Cancellation Request 31

Bit 30 – CR30 Cancellation Request 30

Bit 29 – CR29 Cancellation Request 29

Bit 28 – CR28 Cancellation Request 28

Bit 27 – CR27 Cancellation Request 27

Bit 26 – CR26 Cancellation Request 26

Bit 25 – CR25 Cancellation Request 25

Bit 24 – CR24 Cancellation Request 24

Bit 23 – CR23 Cancellation Request 23

Bit 22 – CR22 Cancellation Request 22

Bit 21 – CR21 Cancellation Request 21

Bit 20 – CR20 Cancellation Request 20

Bit 19 – CR19 Cancellation Request 19

Bit 18 – CR18 Cancellation Request 18

Bit 17 – CR17 Cancellation Request 17

Bit 16 – CR16 Cancellation Request 16

Bit 15 – CR15 Cancellation Request 15

Bit 14 – CR14 Cancellation Request 14

Bit 13 – CR13 Cancellation Request 13

Bit 12 – CR12 Cancellation Request 12

Bit 11 – CR11 Cancellation Request 11

Bit 10 – CR10 Cancellation Request 10

Bit 9 – CR9 Cancellation Request 9

Bit 8 – CR8 Cancellation Request 8

Bit 7 – CR7 Cancellation Request 7

Bit 6 – CR6 Cancellation Request 6

Bit 5 – CR5 Cancellation Request 5

Bit 4 – CR4 Cancellation Request 4

Bit 3 – CR3 Cancellation Request 3

Bit 2 – CR2 Cancellation Request 2

Bit 1 – CR1 Cancellation Request 1

Bit 0 – CR0 Cancellation Request 0

35.7.49 Tx Buffer Transmission Occurred

Name: TXBTO
Offset: 0x1D8
Reset: 0x00000000
Property: R

Table 35-67. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – TO31 Transmission Occurred 31

Bit 30 – TO30 Transmission Occurred 30

Bit 29 – TO29 Transmission Occurred 29

Bit 28 – TO28 Transmission Occurred 28

Bit 27 – TO27 Transmission Occurred 27

Bit 26 – TO26 Transmission Occurred 26

Bit 25 – TO25 Transmission Occurred 25

Bit 24 – TO24 Transmission Occurred 24

Bit 23 – TO23 Transmission Occurred 23

Bit 22 – TO22 Transmission Occurred 22

Bit 21 – TO21 Transmission Occurred 21

Bit 20 – TO20 Transmission Occurred 20

Bit 19 – TO19 Transmission Occurred 19

Bit 18 – TO18 Transmission Occurred 18

Bit 17 – TO17 Transmission Occurred 17

Bit 16 – TO16 Transmission Occurred 16

Bit 15 – TO15 Transmission Occurred 15

Bit 14 – TO14 Transmission Occurred 14

Bit 13 – TO13 Transmission Occurred 13

Bit 12 – TO12 Transmission Occurred 12

Bit 11 – TO11 Transmission Occurred 11

Bit 10 – TO10 Transmission Occurred 10

Bit 9 – TO9 Transmission Occurred 9

Bit 8 – TO8 Transmission Occurred 8

Bit 7 – TO7 Transmission Occurred 7

Bit 6 – TO6 Transmission Occurred 6

Bit 5 – TO5 Transmission Occurred 5

Bit 4 – TO4 Transmission Occurred 4

Bit 3 – TO3 Transmission Occurred 3

Bit 2 – TO2 Transmission Occurred 2

Bit 1 – TO1 Transmission Occurred 1

Bit 0 – TO0 Transmission Occurred 0

35.7.50 Tx Buffer Cancellation Finished

Name: TXBCF
Offset: 0x1DC
Reset: 0x00000000
Property: R

Table 35-68. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – CF31 Tx Buffer Cancellation Finished 31

Bit 30 – CF30 Tx Buffer Cancellation Finished 30

Bit 29 – CF29 Tx Buffer Cancellation Finished 29

Bit 28 – CF28 Tx Buffer Cancellation Finished 28

Bit 27 – CF27 Tx Buffer Cancellation Finished 27

Bit 26 – CF26 Tx Buffer Cancellation Finished 26

Bit 25 – CF25 Tx Buffer Cancellation Finished 25

Bit 24 – CF24 Tx Buffer Cancellation Finished 24

Bit 23 – CF23 Tx Buffer Cancellation Finished 23

Bit 22 – CF22 Tx Buffer Cancellation Finished 22

Bit 21 – CF21 Tx Buffer Cancellation Finished 21
Bit 20 – CF20 Tx Buffer Cancellation Finished 20
Bit 19 – CF19 Tx Buffer Cancellation Finished 19
Bit 18 – CF18 Tx Buffer Cancellation Finished 18
Bit 17 – CF17 Tx Buffer Cancellation Finished 17
Bit 16 – CF16 Tx Buffer Cancellation Finished 16
Bit 15 – CF15 Tx Buffer Cancellation Finished 15
Bit 14 – CF14 Tx Buffer Cancellation Finished 14
Bit 13 – CF13 Tx Buffer Cancellation Finished 13
Bit 12 – CF12 Tx Buffer Cancellation Finished 12
Bit 11 – CF11 Tx Buffer Cancellation Finished 11
Bit 10 – CF10 Tx Buffer Cancellation Finished 10
Bit 9 – CF9 Tx Buffer Cancellation Finished 9
Bit 8 – CF8 Tx Buffer Cancellation Finished 8
Bit 7 – CF7 Tx Buffer Cancellation Finished 7
Bit 6 – CF6 Tx Buffer Cancellation Finished 6
Bit 5 – CF5 Tx Buffer Cancellation Finished 5
Bit 4 – CF4 Tx Buffer Cancellation Finished 4
Bit 3 – CF3 Tx Buffer Cancellation Finished 3
Bit 2 – CF2 Tx Buffer Cancellation Finished 2
Bit 1 – CF1 Tx Buffer Cancellation Finished 1
Bit 0 – CF0 Tx Buffer Cancellation Finished 0

35.7.51 Tx Buffer Transmission Interrupt Enable

Name: TXBTIE
Offset: 0x1E0
Reset: 0x00000000
Property: RW

Table 35-69. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – TIE31 Transmission Interrupt Enable 31

Bit 30 – TIE30 Transmission Interrupt Enable 30

Bit 29 – TIE29 Transmission Interrupt Enable 29

Bit 28 – TIE28 Transmission Interrupt Enable 28

Bit 27 – TIE27 Transmission Interrupt Enable 27

Bit 26 – TIE26 Transmission Interrupt Enable 26

Bit 25 – TIE25 Transmission Interrupt Enable 25

Bit 24 – TIE24 Transmission Interrupt Enable 24

Bit 23 – TIE23 Transmission Interrupt Enable 23

Bit 22 – TIE22 Transmission Interrupt Enable 22

Bit 21 – TIE21 Transmission Interrupt Enable 21
Bit 20 – TIE20 Transmission Interrupt Enable 20
Bit 19 – TIE19 Transmission Interrupt Enable 19
Bit 18 – TIE18 Transmission Interrupt Enable 18
Bit 17 – TIE17 Transmission Interrupt Enable 17
Bit 16 – TIE16 Transmission Interrupt Enable 16
Bit 15 – TIE15 Transmission Interrupt Enable 15
Bit 14 – TIE14 Transmission Interrupt Enable 14
Bit 13 – TIE13 Transmission Interrupt Enable 13
Bit 12 – TIE12 Transmission Interrupt Enable 12
Bit 11 – TIE11 Transmission Interrupt Enable 11
Bit 10 – TIE10 Transmission Interrupt Enable 10
Bit 9 – TIE9 Transmission Interrupt Enable 9
Bit 8 – TIE8 Transmission Interrupt Enable 8
Bit 7 – TIE7 Transmission Interrupt Enable 7
Bit 6 – TIE6 Transmission Interrupt Enable 6
Bit 5 – TIE5 Transmission Interrupt Enable 5
Bit 4 – TIE4 Transmission Interrupt Enable 4
Bit 3 – TIE3 Transmission Interrupt Enable 3
Bit 2 – TIE2 Transmission Interrupt Enable 2
Bit 1 – TIE1 Transmission Interrupt Enable 1
Bit 0 – TIE0 Transmission Interrupt Enable 0

35.7.52 Tx Buffer Cancellation Finished Interrupt Enable

Name: TXBCIE
Offset: 0x1E4
Reset: 0x00000000
Property: RW

Table 35-70. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – CFIE31 Cancellation Finished Interrupt Enable 31

Bit 30 – CFIE30 Cancellation Finished Interrupt Enable 30

Bit 29 – CFIE29 Cancellation Finished Interrupt Enable 29

Bit 28 – CFIE28 Cancellation Finished Interrupt Enable 28

Bit 27 – CFIE27 Cancellation Finished Interrupt Enable 27

Bit 26 – CFIE26 Cancellation Finished Interrupt Enable 26

Bit 25 – CFIE25 Cancellation Finished Interrupt Enable 25

Bit 24 – CFIE24 Cancellation Finished Interrupt Enable 24

Bit 23 – CFIE23 Cancellation Finished Interrupt Enable 23

Bit 22 – CFIE22 Cancellation Finished Interrupt Enable 22

Bit 21 – CFIE21 Cancellation Finished Interrupt Enable 21
Bit 20 – CFIE20 Cancellation Finished Interrupt Enable 20
Bit 19 – CFIE19 Cancellation Finished Interrupt Enable 19
Bit 18 – CFIE18 Cancellation Finished Interrupt Enable 18
Bit 17 – CFIE17 Cancellation Finished Interrupt Enable 17
Bit 16 – CFIE16 Cancellation Finished Interrupt Enable 16
Bit 15 – CFIE15 Cancellation Finished Interrupt Enable 15
Bit 14 – CFIE14 Cancellation Finished Interrupt Enable 14
Bit 13 – CFIE13 Cancellation Finished Interrupt Enable 13
Bit 12 – CFIE12 Cancellation Finished Interrupt Enable 12
Bit 11 – CFIE11 Cancellation Finished Interrupt Enable 11
Bit 10 – CFIE10 Cancellation Finished Interrupt Enable 10
Bit 9 – CFIE9 Cancellation Finished Interrupt Enable 9
Bit 8 – CFIE8 Cancellation Finished Interrupt Enable 8
Bit 7 – CFIE7 Cancellation Finished Interrupt Enable 7
Bit 6 – CFIE6 Cancellation Finished Interrupt Enable 6
Bit 5 – CFIE5 Cancellation Finished Interrupt Enable 5
Bit 4 – CFIE4 Cancellation Finished Interrupt Enable 4
Bit 3 – CFIE3 Cancellation Finished Interrupt Enable 3
Bit 2 – CFIE2 Cancellation Finished Interrupt Enable 2
Bit 1 – CFIE1 Cancellation Finished Interrupt Enable 1
Bit 0 – CFIE0 Cancellation Finished Interrupt Enable 0

35.7.53 Tx Event FIFO Configuration

Name: TXEFC
Offset: 0x1F0
Reset: 0x00000000
Property: RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

Table 35-71. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			EFWM[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
			EFS[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EFSA[13:6]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EFSA[5:0]							
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bits 29:24 – EFWM[5:0] Event FIFO Watermark

Value	Description
0	Watermark interrupt disabled.
1 – 32	Level for Tx Event FIFO watermark interrupt (IR.TEFW bit (IR <13>)).
>32	Watermark interrupt disabled.

Bits 21:16 – EFS[5:0] Event FIFO Size

The Tx Event FIFO elements are indexed from 0 to EFS - 1.

Value	Description
0	Tx Event FIFO disabled
1 – 32	Number of Tx Event FIFO elements.
>32	Values greater than 32 are interpreted as 32.

Bits 15:2 – EFSA[13:0] Event FIFO Start Address

Start address of Tx Event FIFO in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word

addresses, i.e., only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.

35.7.54 Tx Event FIFO Status

Name: TXEFS
Offset: 0x1F4
Reset: 0x00000000
Property: R

Table 35-72. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
							TEFL	EFF
Access							R	R
Reset							0	0

Bit	23	22	21	20	19	18	17	16
				EFPI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
				EFGI[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
			EFFL[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 25 – TEFL Tx Event FIFO Element Lost

This bit is a copy of interrupt flag IR.TEFL bit (IR <15>). When IR.TEFL bit (IR <15>) is reset, this bit is also reset.

Value	Description
0	No Tx Event FIFO element lost.
1	Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.

Bit 24 – EFF Event FIFO Full

Value	Description
0	Tx Event FIFO not full.
1	Tx Event FIFO full.

Bits 20:16 – EFPI[4:0] Event FIFO Put Index

Tx Event FIFO write index pointer, range 0 to 31.

Bits 12:8 – EFGI[4:0] Event FIFO Get Index

Tx Event FIFO read index pointer, range 0 to 31.

Bits 5:0 – EFFL[5:0] Event FIFO Fill Level

Number of elements stored in Tx Event FIFO, range 0 to 32.

35.7.55 Tx Event FIFO Acknowledge

Name: TXEFA
Offset: 0x1F8
Reset: 0x00000000
Property: RW

Table 35-73. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				EFAI[4:0]				
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bits 4:0 – EFAI[4:0] Event FIFO Acknowledge Index

After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI bits (TXEFS <12:8>) to EFAI + 1 and update the FIFO 0 Fill Level TXEFS.EFFL bits (TXEFS <4:0>).

35.7.56 TSU CREL

Name: CREL_TSU
Offset: 0x260
Reset: 0x10000000
Property: R

Table 35-74. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	REL[3:0]				STEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SUBSTEP[3:0]				YEAR[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MON[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DAY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – REL[3:0] Core Release
One digit, BCD-coded.

Bits 27:24 – STEP[3:0] Step of Core Release
One digit, BCD-coded.

Bits 23:20 – SUBSTEP[3:0] Sub-step of Core Release
One digit, BCD-coded.

Bits 19:16 – YEAR[3:0] Timestamp Year
One digit, BCD-coded. This field is set by generic parameter on synthesis.

Bits 15:8 – MON[7:0] Timestamp Month
Two digits, BCD-coded. This field is set by generic parameter on synthesis.

Bits 7:0 – DAY[7:0] Timestamp Day
Two digits, BCD-coded. This field is set by generic parameter on synthesis.

35.7.57 Timestamp Configuration

Name: TSCFG
Offset: 0x264
Reset: 0x00000000
Property: R/W

Table 35-75. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	TBPRES[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
						SCP	TBCS	TSUE
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 15:8 – TBPRES[7:0] Time Base Prescaler

0x00 to 0xFF

Valid values range from 0 to 255. The oscillator frequency is divided by TBPRES + 1 for generating the timebase counter clock.

Bit 2 – SCP Select Capturing Position

Value	Description
0	Capture Timestamp at EOF
1	Capture Timestamp at SOF

Bit 1 – TBCS Timebase Counter Select

Value	Description
0	Timestamp value captured from internal timebase counter, ATB.TB[31:0] is the internal timebase counter.
1	Reserved

Bit 0 – TSUE Timestamp Unit Enable

Value	Description
0	TSU disabled
1	TSU enabled

35.7.58 Timestamp Status 1

Name: TSS1
Offset: 0x268
Reset: 0x00000000
Property: R

Table 35-76. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TSN[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TSN[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TSL[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TSL[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – TSN[15:0] Timestamp New

Each Timestamp register (TS0-TS7) is assigned one bit. The bits are set when a timestamp was stored in the related Timestamp register. Reading a Timestamp register resets the related bit.

Bits 15:0 – TSL[15:0] Timestamp Lost

Each Timestamp register (TS0-TS7) is assigned one bit. The bits are set when the timestamp stored in the related Timestamp register was overwritten before it was read. Reading a Timestamp register resets the related bit.

35.7.59 Timestamp Status 2

Name: TSS2
Offset: 0x26C
Reset: 0x0000A000
Property: R

Table 35-77. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	ITBG[1:0]		NTSG[1:0]					
Access	R	R	R	R				
Reset	1	0	1	0				
Bit	7	6	5	4	3	2	1	0
					TSP[3:0]			
Access					R	R	R	R
Reset					0	0	0	0

Bits 15:14 – ITBG[1:0] Internal Timebase and SOF select Generic
This bitfield is fixed to 0x3.

Value	Description
00	no SOF option, no internal timebase
01	no SOF option, internal timebase (default)
10	SOF option, no internal timebase
11	SOF option, internal timebase

Bits 13:12 – NTSG[1:0] Number of Timestamps Generic
This parameter is fixed to 0x1.

Value	Description
00	4 timestamp registers (TS0-TS3)
01	8 timestamp registers (default, TS0-TS7)
10	16 timestamp registers (TS0-TS15)
11	Reserved

Bits 3:0 – TSP[3:0] Timestamp Pointer
The Timestamp Pointer is incremented by one each time a timestamp is captured. From its maximum value (7), it is incremented to 0.

35.7.60 Timestamp 0

Name: TS0
Offset: 0x270
Reset: 0x00000000
Property: R

Table 35-78. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS0

35.7.61 Timestamp 1

Name: TS1
Offset: 0x274
Reset: 0x00000000
Property: R

Table 35-79. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS1

35.7.62 Timestamp 2

Name: TS2
Offset: 0x278
Reset: 0x00000000
Property: R

Table 35-80. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS2

35.7.63 Timestamp 3

Name: TS3
Offset: 0x27C
Reset: 0x00000000
Property: R

Table 35-81. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS0

35.7.64 Timestamp 4

Name: TS4
Offset: 0x280
Reset: 0x00000000
Property: R

Table 35-82. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS4

35.7.65 Timestamp 5

Name: TS5
Offset: 0x284
Reset: 0x00000000
Property: R

Table 35-83. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS5

35.7.66 Timestamp 6

Name: TS6
Offset: 0x288
Reset: 0x00000000
Property: R

Table 35-84. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS6

35.7.67 Timestamp 7

Name: TS7
Offset: 0x28C
Reset: 0x00000000
Property: R

Table 35-85. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS7

35.7.68 Timestamp 8

Name: TS8
Offset: 0x290
Reset: 0x00000000
Property: R

Table 35-86. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS8

35.7.69 Timestamp 9

Name: TS9
Offset: 0x294
Reset: 0x00000000
Property: R

Table 35-87. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS9

35.7.70 Timestamp 10

Name: TS10
Offset: 0x298
Reset: 0x00000000
Property: R

Table 35-88. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS10

35.7.71 Timestamp 11

Name: TS11
Offset: 0x29C
Reset: 0x00000000
Property: R

Table 35-89. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS11

35.7.72 Timestamp 12

Name: TS12
Offset: 0x2A0
Reset: 0x00000000
Property: R

Table 35-90. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS12

35.7.73 Timestamp 13

Name: TS13
Offset: 0x2A4
Reset: 0x00000000
Property: R

Table 35-91. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS13

35.7.74 Timestamp 14

Name: TS14
Offset: 0x2A8
Reset: 0x00000000
Property: R

Table 35-92. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS14

35.7.75 Timestamp 15

Name: TS15
Offset: 0x2AC
Reset: 0x00000000
Property: R

Table 35-93. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TS[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TS[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TS[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TS[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TS[31:0] Timestamp Word TS15

35.7.76 Actual Timebase

Name: ATB
Offset: 0x2B0
Reset: 0x00000000
Property: R/K/C

Table 35-94. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TB[31:24]							
Access	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TB[23:16]							
Access	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TB[15:8]							
Access	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TB[7:0]							
Access	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C	R/K/C
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TB[31:0] Timebase for timestamp generation

35.8 Rx Buffer and FIFO Element Register Summary

35.8.1 Rx Buffer and FIFO Element 0

Name: R0
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 35-95. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ESI	XTD	RTR	ID[28:24]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ID[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ID[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – ESI Error State Indicator

Bit 30 – XTD Extended Identifier

Bit 29 – RTR Remote Transmission Request

Bits 28:0 – ID[28:0] Identifier

35.8.2 Rx Buffer and FIFO Element 1

Name: R1
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 35-96. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ANMF	FIDX[6:0]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
			FDF	BRS	DLC[3:0]			
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	RXTS[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RXTS[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – ANMF Accepted Non-matching Frame

Bits 30:24 – FIDX[6:0] Filter Index

Bit 21 – FDF FD Format

Bit 20 – BRS Bit Rate Search

Bits 19:16 – DLC[3:0] Data Length Code

Bits 15:0 – RXTS[15:0] Rx Timestamp

35.8.3 Rx Buffer and FIFO Element Data

Name: R2
Offset: 0x08 + n*0x04 [n=0..15]
Reset: 0x00000000
Property: RW

Table 35-97. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DB3[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DB2[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DB1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DB0[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DB3[7:0] Data Byte 3

Bits 23:16 – DB2[7:0] Data Byte 2

Bits 15:8 – DB1[7:0] Data Byte 1

Bits 7:0 – DB0[7:0] Data Byte 0

35.9 Tx Buffer Element Register Summary

35.9.1 Tx Buffer Element 0

Name: T0
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 35-98. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ESI	XTD	RTR	ID[28:24]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ID[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ID[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – ESI Error State Indicator

Bit 30 – XTD Extended Identifier

Bit 29 – RTR Remote Transmission Request

Bits 28:0 – ID[28:0] Identifier

35.9.2 Tx Buffer Element 1

Name: T1
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 35-99. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MML[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EFC	TSCE	FDF	BRS	DLC[3:0]			
Access	RW	R/W	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MMH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:24 – MML[7:0] Message Marker Lower Byte

Bit 23 – EFC Event FIFO Control

Bit 22 – TSCE Time Stamp Capture Enable for TSU

Bit 21 – FDF FD Format

Bit 20 – BRS Bit Rate Search

Bits 19:16 – DLC[3:0] Identifier

Bits 15:8 – MMH[7:0] Message Marker Higher Byte

35.9.3 Tx Buffer Element Data

Name: T2
Offset: 0x08 + n*0x04 [n=0..15]
Reset: 0x00000000
Property: RW

Table 35-100. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DB3[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DB2[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DB1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DB0[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – DB3[7:0] Data Byte 3

Bits 23:16 – DB2[7:0] Data Byte 2

Bits 15:8 – DB1[7:0] Data Byte 1

Bits 7:0 – DB0[7:0] Data Byte 0

35.10 Tx Event FIFO Element Register Summary

35.10.1 Tx Event FIFO Element 0

Name: E0
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 35-101. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ESI	XTD	RTR	ID[28:24]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ID[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	ID[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – ESI Error State Indicator

Bit 30 – XTD Extended Identifier

Bit 29 – RTR Remote Transmission Request

Bits 28:0 – ID[28:0] Identifier

35.10.2 Tx Event FIFO Element 1

Name: E1
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 35-102. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MM[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ET[1:0]		FDF	BRS	DLC[3:0]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TXTS[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TXTS[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – MM[7:0] Message Marker

Bits 23:22 – ET[1:0] Event Type

Value	Name	Description
1	TXE	Tx event
2	TXC	Transmission in spite of cancellation

Bit 21 – FDF FD Format

Bit 20 – BRS Bit Rate Search

Bits 19:16 – DLC[3:0] Data Length Code

Bits 15:0 – TXTS[15:0] Tx Timestamp

35.11 Standard Message ID Filter Element Register Summary

35.11.1 Standard Message ID Filter Element

Name: S0
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 35-103. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SFT[1:0]		SFEC[2:0]			SFID1[10:8]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SFID1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						SFID2[10:8]		
Access						RW	RW	RW
Reset						0	0	0
Bit	7	6	5	4	3	2	1	0
	SFID2[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – SFT[1:0] Standard Filter Type

Value	Name	Description
0	RANGE	Range filter from SFID1 to SFID2
1	DUAL	Dual ID filter for SFID1 or SFID2
2	CLASSIC	Classic filter

Bits 29:27 – SFEC[2:0] Standard Filter Element Configuration

Value	Name	Description
0	DISABLE	Disable filter element
1	STF0M	Store in Rx FIFO 0 if filter match
2	STF1M	Store in Rx FIFO 1 if filter match
3	REJECT	Reject ID if filter match
4	PRIORITY	Set priority if filter match
5	PRIF0M	Set priority and store in FIFO 0 if filter match
6	PRIF1M	Set priority and store in FIFO 1 if filter match
7	STRXBUF	Store into Rx Buffer

Bits 26:16 – SFID1[10:0] Standard Filter ID 1

Bits 10:0 – SFID2[10:0] Standard Filter ID 2

35.12 Extended Message ID Filter Element Register Summary

35.12.1 Extended Message ID Filter Element 0

Name: F0
Offset: 0x000
Reset: 0x00000000
Property: RW

Table 35-104. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EFEC[2:0]			EFID1[28:24]				
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EFID1[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EFID1[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EFID1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:29 – EFEC[2:0] Extended Filter Element Configuration

Value	Name	Description
0	DISABLE	Disable filter element
1	STF0M	Store in Rx FIFO 0 if filter match
2	STF1M	Store in Rx FIFO 1 if filter match
3	REJECT	Reject ID if filter match
4	PRIORITY	Set priority if filter match
5	PRIF0M	Set priority and store in FIFO 0 if filter match
6	PRIF1M	Set priority and store in FIFO 1 if filter match
7	STRXBUF	Store into Rx Buffer

Bits 28:0 – EFID1[28:0] Extended Filter ID 1

35.12.2 Extended Message ID Filter Element 1

Name: F1
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 35-105. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	EFT[1:0]			EFID2[28:24]				
Access	RW	RW		RW	RW	RW	RW	RW
Reset	0	0		0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	EFID2[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EFID2[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EFID2[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:30 – EFT[1:0] Extended Filter Type

Value	Name	Description
0	RANGEM	Range filter from EFID1 to EFID2
1	DUAL	Dual ID filter for EFID1 or EFID2
2	CLASSIC	Classic filter
3	RANGE	Range filter from EFID1 to EFID2 with no XIDAM mask

Bits 28:0 – EFID2[28:0] Extended Filter ID 2

36. Universal Serial Bus (USB)

36.1 Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting device modes.

The USB device mode supports 8 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 16 endpoints. Each endpoint is fully configurable in any of the four transfer types: control, interrupt, bulk or isochronous. The maximum data payload size is selectable up to 1023 bytes.

Internal SRAM is used to keep the configuration and data buffer for each endpoint. The memory locations used for the endpoint configurations and data buffers is fully configurable. The amount of memory allocated is dynamic according to the number of endpoints in use, and the configuration of these. The USB module has a built-in Direct Memory Access (DMA) and will read/write data from/to the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

To maximize throughput, an endpoint can be configured for ping-pong operation. When this is done the input and output endpoint with the same address are used in the same direction. The CPU or DMA Controller can then read/write one data buffer while the USB module writes/reads from the other buffer. This gives double buffered communication.

Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without any software intervention. This reduces the number of interrupts and software intervention needed for USB transfers.

For low power operation the USB module can put the microcontroller in any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resume, the USB module can wake the microcontroller from any sleep mode.

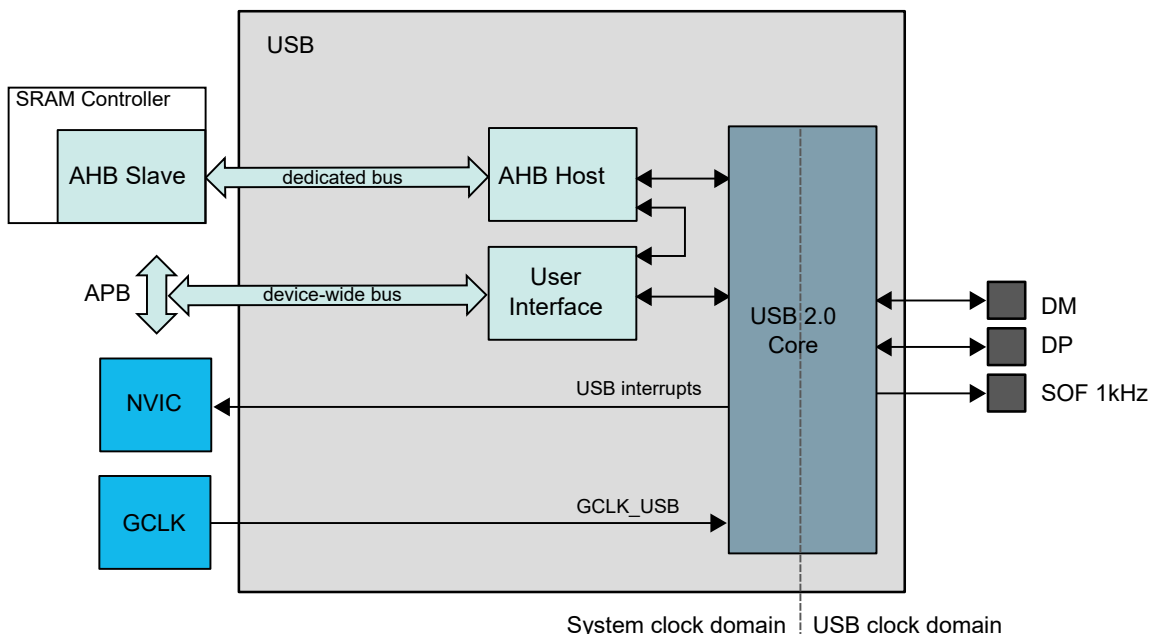
36.2 Features

- Compatible with the USB 2.1 specification
- USB Host and Device mode
- Supports full (12 Mbit/s) and low (1.5 Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1 kHz SOF clock available on external pin
- Device mode
 - Supports 8 IN endpoints and 8 OUT endpoints
 - No endpoint size limitations
 - Built-in DMA with multi-packet and dual bank for all endpoints
 - Supports feedback endpoint
 - Supports crystal less clock
- Host mode
 - Supports 8 physical pipes
 - No pipe size limitations
 - Supports multiplexed virtual pipe on one physical pipe to allow an unlimited USB tree
 - Built-in DMA with multi-packet support and dual bank for all pipes

- Supports feedback endpoint
- Supports the USB 2.0 Phase-locked SOFs feature

36.3 USB Block Diagram

Figure 36-1. LS/FS Implementation: USB Block Diagram



36.4 Signal Description

Pin Name	Pin Description	Type
DM	Data -: Differential Data Line - Port	Input/Output
DP	Data +: Differential Data Line + Port	Input/Output
SOF 1 kHz	SOF Output	Output

Note: A 1 kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1 kHz SOF clock to the peripheral function. The SOF clock is available for device and host mode.

Refer to the [Pinout](#) for details on the pin mapping for this peripheral.

36.5 Functional Description

36.5.1 USB General Operation

36.5.1.1 Initialization

The USB module requires a GCLK_USB of 48 MHz $\pm$ 0.25% clock for low-speed and full-speed operation.

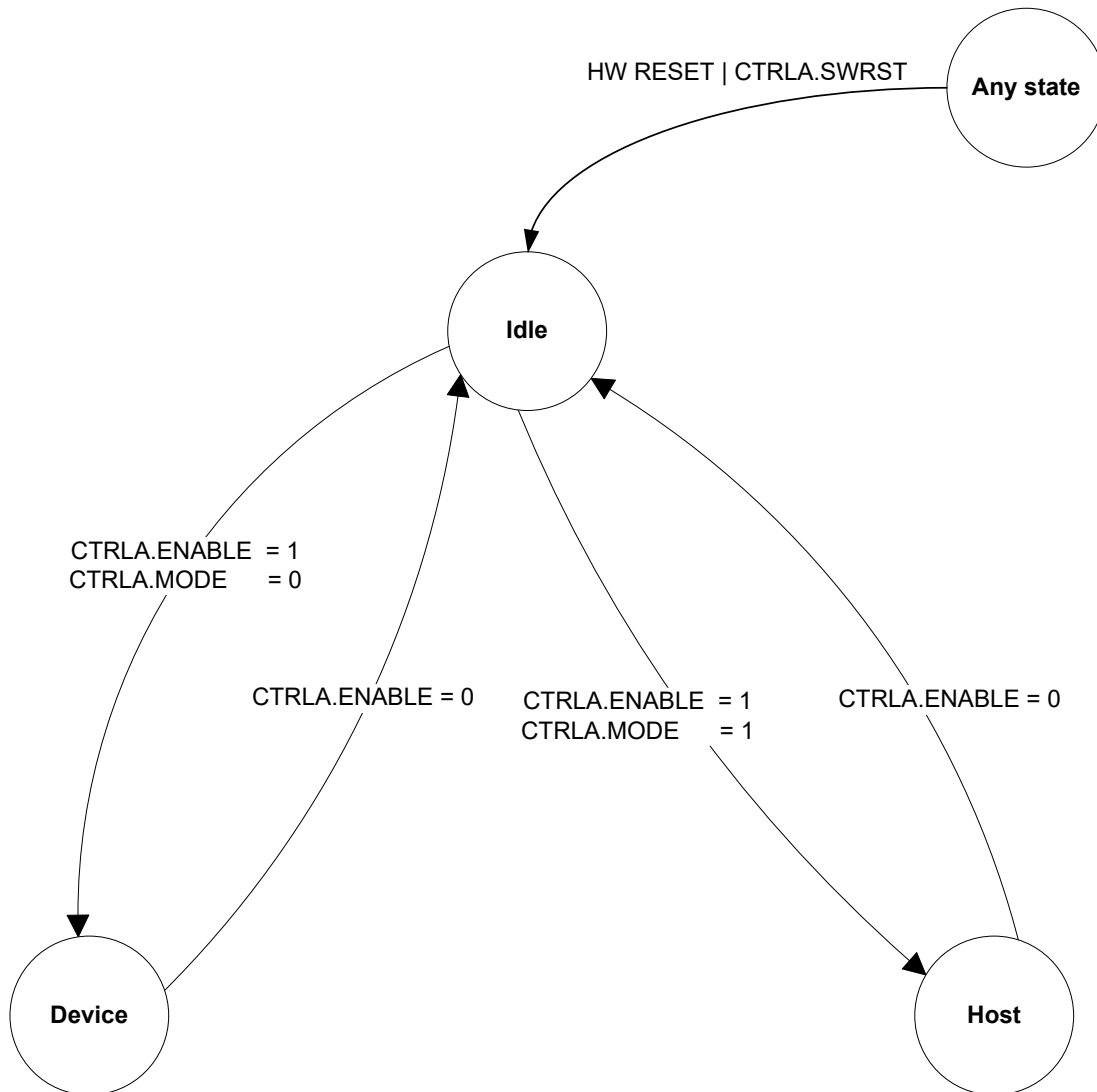
Clock recovery is achieved by a digital Phase-Locked Loop in the USB module, which complies with the USB jitter specifications. If crystal-less operation is used in USB device mode, for additional information, refer to the *USB Clock Recovery Module*.

After a hardware reset, the USB is in the idle state. In this state:

- The module is disabled. The USB Enable bit in the Control A register (CTRLA.ENABLE) is reset.
- The module clock is stopped in order to minimize power consumption.

- The USB pad is in suspend mode.
- The internal states and registers of the device are reset.

Figure 36-2. General States



The output drivers for the DP/DM USB line interface can be fine tuned with calibration values from production tests. The calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register (PADCAL) by software, before enabling the USB, to achieve the specified accuracy. Refer to the *NVM Software Calibration Area Mapping* for additional information.

For additional information, refer to the Pad Calibration (PADCAL) register.

The USB is enabled by writing a '1' to CTRLA.ENABLE. The USB is disabled by writing a '0' to CTRLA.ENABLE.

The USB is reset by writing a '1' to the Software Reset bit in CTRLA (CTRLA.SWRST). All registers in the USB will be reset to their initial state, and the USB will be disabled. Refer to the Control A (CTRLA) register for details.

The user can configure pads and speed before enabling the USB by writing to the Operating Mode bit in the Control A register (CTRLA.MODE) and the Speed Configuration field in the Control B

register (CTRLB.SPDCONF). These values are taken into account once the USB has been enabled by writing a '1' to CTRLA.ENABLE.

After writing a '1' to CTRLA.ENABLE, the USB enters device mode or host mode (according to CTRLA.MODE).

The USB can be disabled at any time by writing a '0' to CTRLA.ENABLE.

Refer to [USB Device Operations](#) for the basic operation of the device mode.

Refer to [Host Operations](#) for the basic operation of the host mode.

36.5.1.2 Debug Operation

When the CPU is halted in debug mode the USB peripheral continues normal operation. If the USB peripheral is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

36.5.2 USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

36.5.2.1 Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

36.5.2.2 Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB host (built-in DMA) with the help of the endpoint descriptors. The base address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer to the section “Endpoint Descriptor Structure”.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

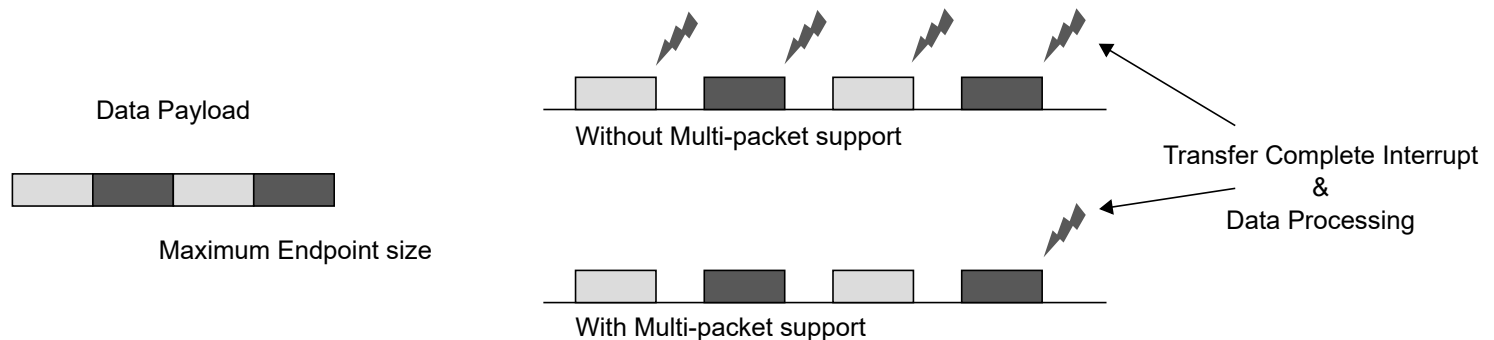
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

36.5.2.3 Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.

Figure 36-3. Multi-Packet Feature - Reduction of CPU Overhead



36.5.2.4 USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

36.5.2.5 Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

36.5.2.6 Management of SETUP Transactions

When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.

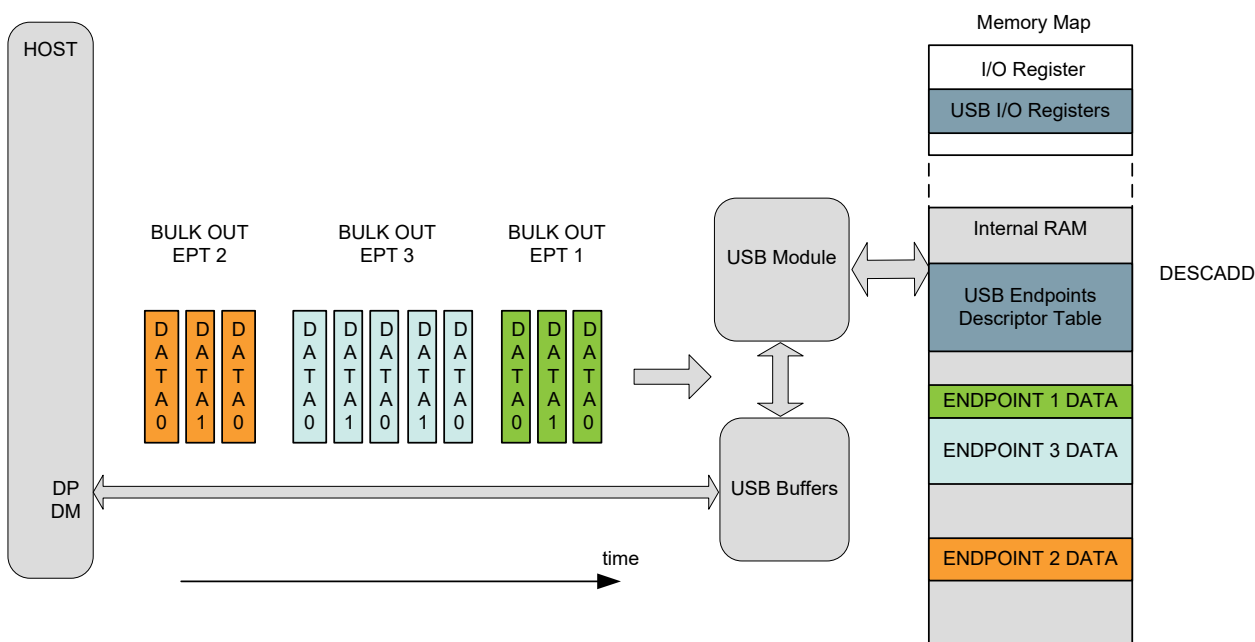
When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. Software must never report a endpoint size to the host that is greater than the value configured in PCKSIZE.SIZE. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If data is successfully received, an ACK handshake is returned to the host, and the number of received data bytes, excluding the CRC, is written to the Byte Count (PCKSIZE.BYTE_COUNT). If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE, no CRC data is written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally the EPSTATUS is updated. Data Toggle OUT bit (EPSTATUS.DTGLOUT), the Data Toggle IN bit (EPSTATUS.DTGLIN), the current bank bit (EPSTATUS.CURRBK) and the Bank Ready 0 bit (EPSTATUS.BK0RDY) are set. Bank Ready 1 bit (EPSTATUS.BK1RDY) and the Stall Bank 0/1 bit (EPSTATUS.STALLQR0/1) are cleared on receiving the SETUP request. The RXSTP bit is set and triggers an interrupt if the Received Setup Interrupt Enable bit is set in Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

36.5.2.7 Management of OUT Transactions

Figure 36-4. OUT Transfer: Data Packet Host to USB Device



When an OUT token is detected, and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

If the address matches, the USB module checks if the endpoint number received is enabled in the EPCFG of the addressed endpoint. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks the Endpoint Configuration register (EPCFG) of the addressed output endpoint. If the type of the endpoint (EPCFG.EPTYPE0) is not set to OUT, the USB module returns to idle and waits for the next token packet.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor, and waits for a DATA0 or DATA1 packet. If a PID error or any other PID than DATA0 or DATA1 is detected, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ0 in EPSTATUS is set, the incoming data is discarded. If the endpoint is not isochronous, a STALL handshake is returned to the host and the Transmit Stall Bank 0 interrupt bit in EPINTFLAG (EPINTFLAG.STALL0) is set.

For isochronous endpoints, data from both a DATA0 and DATA1 packet will be accepted. For other endpoint types the PID is checked against EPSTATUS.DTGLOUT. If a PID mismatch occurs, the incoming data is discarded, and an ACK handshake is returned to the host.

If EPSTATUS.BKORDY is set, the incoming data is discarded, the bit Transmit Fail 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRFAIL0) and the status bit STATUS_BK.ERRORFLOW are set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the maximum data payload specified as PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If the endpoint is isochronous and a bit-stuff or CRC error in the incoming data, the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. Finally the EPINTFLAG.TRFAIL0 and CRC Error bit in the Device Bank Status register (STATUS_BK.CRCERR) is set for the addressed endpoint.

If data was successfully received, an ACK handshake is returned to the host if the endpoint is not isochronous, and the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE no CRC data bytes are written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data byte is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally in EPSTATUS for the addressed output endpoint, EPSTATUS.BK0RDY is set and EPSTATUS.DTGLOUT is toggled if the endpoint is not isochronous. The flag Transmit Complete 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRCPT0) is set for the addressed endpoint.

36.5.2.8 Multi-Packet Transfers for OUT Endpoint

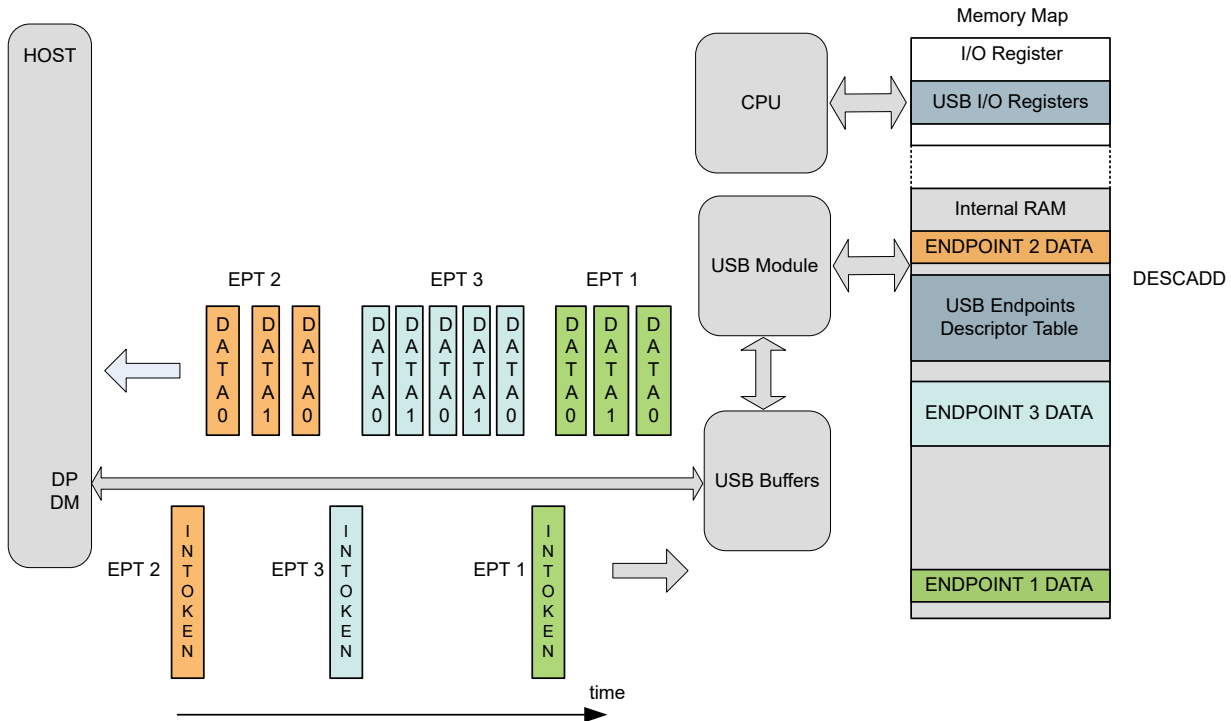
The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. This value must be a multiple of PCKSIZE.SIZE, otherwise excess data may be written to SRAM locations used by other parts of the application.

EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

If a maximum payload size packet is received, PCKSIZE.BYTE_COUNT will be incremented by PCKSIZE.SIZE after the transaction has completed, and EPSTATUS.DTGLOUT will be toggled if the endpoint is not isochronous. If the updated PCKSIZE.BYTE_COUNT is equal to PCKSIZE.MULTI_PACKET_SIZE (i.e. the last transaction), EPSTATUS.BK1RDY/BK0RDY, and EPINTFLAG.TRCPT0/TRCPT1 will be set.

36.5.2.9 Management of IN Transactions

Figure 36-5. IN Transfer: Data Packet USB Device to Host After Request from Host



When an IN token is detected, and if the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint received is enabled in the EPCFG of the addressed endpoint and if not, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed input endpoint. If the EPCFG.EPTYPE1 is not set to IN, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ1 in EPSTATUS is set, and the endpoint is not isochronous, a STALL handshake is returned to the host and EPINTFLAG.STALL1 is set.

If EPSTATUS.BK1RDY is cleared, the flag EPINTFLAG.TRFAIL1 is set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor. The data pointed to by the Data Buffer Address (ADDR) is sent to the host in a DATA0 packet if the endpoint is isochronous. For non-isochronous endpoints a DATA0 or DATA1 packet is sent depending on the state of EPSTATUS.DTGLIN. When the number of data bytes specified in endpoint PCKSIZE.BYTE_COUNT is sent, the CRC is appended and sent to the host.

For isochronous endpoints, EPSTATUS.BK1RDY is cleared and EPINTFLAG.TRCPT1 is set.

For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

36.5.2.10 Multi-Packet Transfers for IN Endpoint

The total number of data bytes to be sent is written to PCKSIZE.BYTE_COUNT as for normal operation. The Multi-packet size register (PCKSIZE.MULTI_PACKET_SIZE) is used to store the number of bytes that are sent, and must be written to zero when setting up a new transfer.

When an IN token is received, PCKSIZE.BYTE_COUNT and PCKSIZE.MULTI_PACKET_SIZE are fetched. If PCKSIZE.BYTE_COUNT minus PCKSIZE.MULTI_PACKET_SIZE is less than the endpoint PCKSIZE.SIZE, endpoint BYTE_COUNT minus endpoint PCKSIZE.MULTI_PACKET_SIZE bytes are transmitted, otherwise PCKSIZE.SIZE number of bytes are transmitted. If endpoint PCKSIZE.BYTE_COUNT is a multiple of PCKSIZE.SIZE, the last packet sent will be zero-length if the AUTOZLP bit is set.

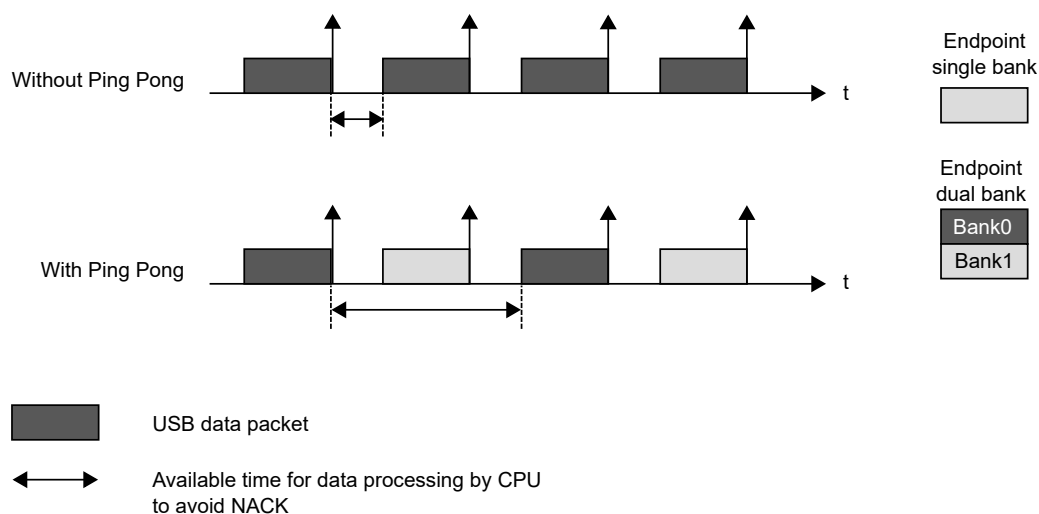
If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTGLIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), MULTI_PACKET_SIZE is incremented by the data payload. EPSTATUS.BK0/1RDY will be cleared and EPINTFLAG.TRCPT0/1 will be set.

36.5.2.11 Ping-Pong Operation

When an endpoint is configured for ping-pong operation, it uses both the input and output data buffers (banks) for a given endpoint in a single direction. The direction is selected by enabling one of the IN or OUT direction in EPCFG.EPTYPE0/1 and configuring the opposite direction in EPCFG.EPTYPE1/0 as Dual Bank.

When ping-pong operation is enabled for an endpoint, the endpoint in the opposite direction must be configured as dual bank. The data buffer, data address pointer and byte counter from the enabled endpoint are used as Bank 0, while the matching registers from the disabled endpoint are used as Bank 1.

Figure 36-6. Ping-Pong Overview



The Bank Select flag in EPSTATUS.CURBK indicates which bank data will be used in the next transaction, and is updated after each transaction. According to EPSTATUS.CURBK, EPINTFLAG.TRCPT0 or EPINTFLAG.TRFAL0 or EPINTFLAG.TRCPT1 or EPINTFLAG.TRFAL1 in

EPINTFLAG and Data Buffer 0/1 ready (EPSTATUS.BK0RDY and EPSTATUS.BK1RDY) are set. The EPSTATUS.DTGLOUT and EPSTATUS.DTGLIN are updated for the enabled endpoint direction only.

36.5.2.12 Feedback Operation

Feedback endpoints are endpoints with the same address but in different directions. This is usually used in explicit feedback mechanism in USB Audio, where a feedback endpoint is associated to one or more isochronous data endpoints to which it provides feedback service.

The feedback endpoint always has the opposite direction from the data endpoint(s). The feedback endpoint has the same endpoint number as the first (lower) data endpoint. A feedback endpoint can be created by configuring an endpoint with different endpoint size (PCKSIZE.SIZE) and different endpoint type (EPCFG.EPTYPE0/1) for the IN and OUT direction.

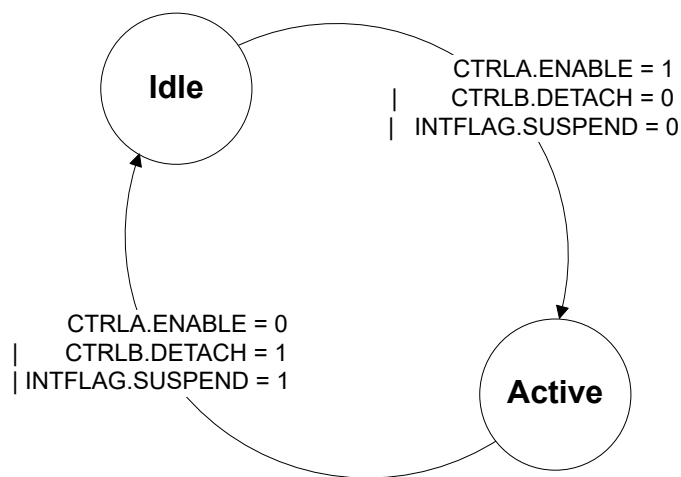
Example Configuration for Feedback Operation:

- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0 = Isochronous OUT, PCKSIZE.SIZE = 512.

36.5.2.13 Suspend State and Pad Behavior

The following figure, Pad Behavior, illustrates the behavior of the USB pad in Device mode.

Figure 36-7. Pad Behavior

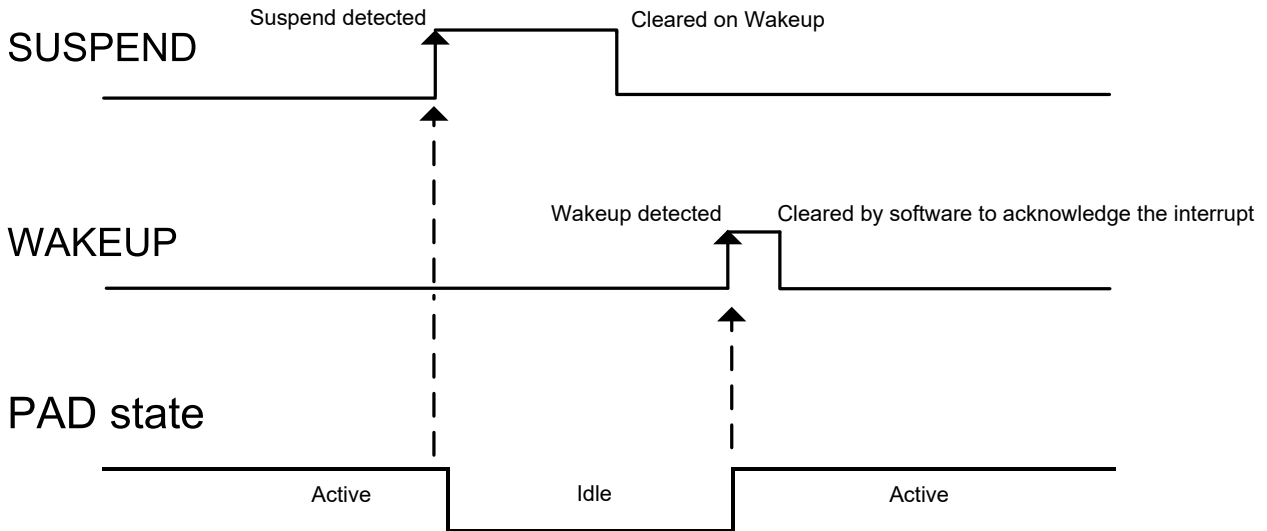


In Idle state, the pad is in Low Power Consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.

Figure 36-8. Pad Events



The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit (INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

36.5.2.14 Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a Suspend state on the bus, that is, the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit (CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit (INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.

In the case where the CTRLB.UPRSM bit is set while a host initiated downstream resume is already started, the CTRLB.UPRSM is cleared and the upstream resume request is ignored.

36.5.2.15 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Device

The LPM Handshake bit in CTRLB.LPMHDSK should be configured to accept the LPM transaction.

When a LPM transaction is received on any enabled endpoint 'n' and a handshake has been sent in response by the controller according to CTRLB.LPMHDSK, the Device Link Power Manager (EXTREG) register is updated in the bank 0 of the addressed endpoint's descriptor. It contains information, such as the Best Effort Service Latency (BESL), the Remote Wake bit (bRemoteWake), and the Link State parameter (bLinkState). Usually, the LPM transaction uses only the endpoint number 0.

If the LPM transaction was positively acknowledged (ACK handshake), USB sets the Link Power Management Interrupt bit (INTFLAG.LPMSUSP) which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (see the BESL parameter in the Extended Register (EXTREG)).

Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section Suspend State and Pad Behavior) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag (INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with no handshake, no flag is set, and the transaction is ignored.

36.5.3 Host Operations

This section gives an overview of the USB module Host operation during normal transactions. For more details on general USB and USB protocol, refer to Universal Serial Bus Specification revision 2.1.

36.5.3.1 Device Detection and Disconnection

Prior to device detection, the software must set the VBUS is OK bit (CTRLB.VBUSOK) in the CTRLB register when the VBUS is available. This notifies the USB host that USB operations can be started. When the CTRLB.VBUSOK bit is '0' and even if the USB HOST is configured and enabled, host operation is halted. Setting the CTRLB.VBUSOK bit will allow host operation when the USB is configured.

The Device detection is managed by the software using the Line State field in the Host Status (STATUS.LINESTATE) register. The device connection is detected by the host controller when DP or DM is pulled high, depending of the speed of the device.

The device disconnection is detected by the host controller when both DP and DM are pulled down using the STATUS.LINESTATE registers.

The Device Connection Interrupt bit (INTFLAG.DCONN) is set if a device connection is detected.

The Device Disconnection Interrupt bit (INTFLAG.DDISC) is set if a device disconnection is detected.

36.5.3.2 Host Terminology

In host mode, the term pipe is used instead of endpoint. A host pipe corresponds to a device endpoint, refer to "Universal Serial Bus Specification revision 2.1." for more information.

36.5.3.3 USB Reset

The USB sends a USB reset signal when the user writes a one to the USB Reset bit (CTRLB.BUSRESET). When the USB reset has been sent, the USB Reset Sent Interrupt bit in the INTFLAG (INTFLAG.RST) is set and all pipes will be disabled.

If the bus was previously in a suspended state (i.e., the Start of Frame Generation Enable bit (CTRLB.SOFE) is zero), the USB will switch it to the Resume state, causing the bus to asynchronously set the Host Wakeup Interrupt flag (INTFLAG.WAKEUP). The CTRLB.SOFE bit will be set in order to generate SOFs immediately after the USB reset.

During USB reset the following registers are cleared:

- The All Host Pipe Configuration register (PCFG)
- The Host Frame Number register (FNUM)
- The Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- The Host Start-of-Frame Control register (HSOFC)
- The Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- The Pipe Interrupt Flag register (PINTFLAG)
- The Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

After the reset, the user should check the Speed Status field in the Status register (STATUS.SPEED) to find out the current speed according to the capability of the peripheral.

36.5.3.4 Pipe Configuration

Pipe data can be placed anywhere in the RAM. The USB controller accesses these pipes directly through the AHB host (built-in DMA) with the help of the pipe descriptors. The base address of the pipe descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer to the section *Pipe Descriptor Structure*.

Before using a pipe, the user should configure the direction and type of the pipe in the Type of Pipe field in the Host Pipe Configuration register (PCFG.PTYPE). The pipe descriptor registers should be initialized to known values before using the pipe, so that the USB controller does not read the random values from the RAM.

The Pipe Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported by the device for the endpoint associated with this pipe. The Address of Data Buffer register (ADDR) should be set to the data buffer used for pipe transfers.

The Pipe Bank bit (PCFG.BK) should be set to '1' if dual banking is desired. Dual bank is not supported for Control pipes.

The Ram Access Interrupt bit in Host Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during an OUT stage.

When a pipe is disabled, the following registers are cleared for that pipe:

- The Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- The Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- The Pipe Interrupt Flag register (PINTFLAG)
- The Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

36.5.3.5 Pipe Activation

A disabled pipe is inactive, and will be reset along with its context registers (pipe registers for the pipe n). Pipes are enabled by writing the Type of the Pipe bit (PCFG.PTYPE) to a value different than 0x0 (disabled).

When a pipe is enabled, the Pipe Freeze bit in the Pipe Status register (PSTATUS.FREEZE) is set. This allows the user to complete the configuration of the pipe, without starting a USB transfer.

When starting an enumeration, the user retrieves the device descriptor by sending a GET_DESCRIPTOR USB request. This descriptor contains the maximal packet size of the device default control endpoint (bMaxPacketSize0), which the user should use to reconfigure the size of the default control pipe.

36.5.3.6 Pipe Address Setup

Once the device has answered the first host requests with the default device address 0, the host assigns a new address to the device. The host controller has to send a USB reset to the device and a SET_ADDRESS(addr) SETUP request with the new address to be used by the device. Once this SETUP transaction is complete, the user writes the new address to the Pipe Device Address field in the Host

Control Pipe register (CTRL_PIPE.PDADDR) in Pipe descriptor. All following requests by this pipe will be performed using this new address.

36.5.3.7 Suspend and Wakeup

Setting CTRLB.SOFE to zero when in host mode will cause the USB to cease sending Start-of-Frames on the USB bus and enter the Suspend state. The USB device will enter the Suspend state 3ms later.

Before entering suspend by writing CTRLB.SOFE to zero, the user must freeze the active pipes by setting their PSTATUS.FREEZE bit. Any current on-going pipe will complete its transaction, and then all pipes will be inactive. The user should wait at least 1 complete frame before entering the suspend mode to avoid any data loss.

The device can awaken the host by sending an Upstream Resume (Remote Wakeup feature). When the host detects a non-idle state on the USB bus, it sets the INTFLAG.WAKEUP. If the non-idle bus state corresponds to an Upstream Resume (K state), the Upstream Resume Received Interrupt bit in the INTFLAG register (INTFLAG.UPRSM) is set and the user must generate a Downstream Resume within 1 ms and for at least 20 ms. It is required to first write a one to the Send USB Resume bit in the CTRLB register (CTRLB.RESUME) to respond to the upstream resume with a downstream resume. Alternatively, the host can resume from a suspend state by sending a Downstream Resume on the USB bus (CTRLB.RESUME set to 1). In both cases, when the downstream resume is completed, the CTRLB.SOFE bit is automatically set and the host enters again the active state.

36.5.3.8 Phase-locked SOFs

To support the Synchronous Endpoints capability, the period of the emitted Start-of-Frame is maintained while the USB connection is not in the active state. This does not apply for the disconnected/connected/reset states. It applies for active, idle, suspend, and resume states. The period of Start-of-Frame will be 1ms when the USB connection is in active state and an integer number of milli-seconds across idle, suspend, and resume states.

To ensure the Synchronous Endpoints capability, the GCLK_USB clock must be kept running. If the GCLK_USB is interrupted, the period of the emitted Start-of-Frame will be erratic.

36.5.3.9 Management of Control Pipes

A control transaction is composed of three stages:

- SETUP
- Data (IN or OUT)
- Status (IN or OUT)

The user has to change the pipe token according to each stage using the Pipe Token field in PCFG (PCFG.PTOKEN).

For control pipes only, the token is assigned a specific initial data toggle sequence:

- SETUP: Data0
- IN: Data1
- OUT: Data1

36.5.3.10 Management of IN Pipes

IN packets are sent by the USB device controller upon IN request reception from the host. All the received data from the device to the host will be stored in the bank provided the bank is empty. The pipe and its descriptor in RAM must be configured.

The host indicates it is able to receive data from the device by clearing the Bank 0/1 Ready bit in PSTATUS (PSTATUS.BK0/1RDY), which means that the memory for the bank is available for new USB transfer.

The USB will perform IN requests as long as the pipe is not frozen by the user.

The generation of IN requests starts when the pipe is unfrozen (PSTATUS.PFREEZE is set to zero).

When the current bank is full, the Transmit Complete 0/1 bit in PINTFLAG (PINTFLAG.TRCPT0/1) will be set and trigger an interrupt if enabled and the PSTATUS.BK0/1RDY bit will be set.

PINTFLAG.TRCPT0/1 must be cleared by software to acknowledge the interrupt. This is done by writing a one to the PINTFLAG.TRCPT0/1 of the addressed pipe.

The user reads the PCKSIZE.BYTE_COUNT to know how many bytes should be read.

To free the bank the user must read the IN data from the address ADDR in the pipe descriptor and clear the PKSTATUS.BK0/1RDY bit. When the IN pipe is composed of multiple banks, a successful IN transaction will switch to the next bank. Another IN request will be performed by the host as long as the PSTATUS.BK0/1RDY bit for that bank is set. The PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1RDY will be updated accordingly.

The user can follow the current bank looking at Current Bank bit in PSTATUS (PSTATUS.CURBK) and by looking at Data Toggle for IN pipe bit in PSTATUS (PSTATUS.DTGLIN).

When the pipe is configured as single bank (Pipe Bank bit in PCFG (PCFG.BK) is 0), only PINTFLAG.TRCPT0 and PSTATUS.BK0 are used. When the pipe is configured as dual bank (PCFG.BK is 1), both PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1 are used.

36.5.3.11 Management of OUT Pipes

OUT packets are sent by the host. All the data stored in the bank will be sent to the device provided the bank is filled. The pipe and its descriptor in RAM must be configured.

The host can send data to the device by writing to the data bank 0 in single bank or the data bank 0/1 in dual bank.

The generation of OUT packet starts when the pipe is unfrozen (PSTATUS.PFREEZE is zero).

The user writes the OUT data to the data buffer pointer by ADDR in the pipe descriptor and allows the USB to send the data by writing a one to the PSTATUS.BK0/1RDY. This will also cause a switch to the next bank if the OUT pipe is part of a dual bank configuration.

PINTFLAGn.TRCPT0/1 must be cleared before setting PSTATUS.BK0/1RDY to avoid missing an PINTFLAGn.TRCPT0/1 event.

36.5.3.12 Alternate Pipe

The user has the possibility to run sequentially several logical pipes on the same physical pipe. It allows addressing of any device endpoint of any attached device on the bus.

Before switching pipe, the user should save the pipe context (Pipe registers and descriptor for pipe n).

After switching pipe, the user should restore the pipe context (Pipe registers and descriptor for pipe n) and in particular PCFG, and PSTATUS.

36.5.3.13 Data Flow Error

This error exists only for isochronous and interrupt pipes for both IN and OUT directions. It sets the Transmit Fail bit in PINTFLAG (PINTFLAG.TRFAIL), which triggers an interrupt if the Transmit Fail bit in PINTENCLR/SET (PINTENCLR/SET.TRFAIL) is set. The user must check the Pipe Interrupt Summary register (PINTSMRY) to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the Pipe Bank Status register (STATUS_BK) for each bank. If the Error Flow bit in the STATUS_BK (STATUS_BK.ERRORFLOW) is set then the user is able to determine the origin of the data flow error. As the user knows that the endpoint is an IN or OUT the error flow can be deduced as OUT underflow or as an IN overflow.

An underflow can occur during an OUT stage if the host attempts to send data from an empty bank. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

An overflow can occur during an IN stage if the device tries to send a packet while the bank is full. Typically this occurs when a CPU is not fast enough. The packet data is not written to the bank and is

lost. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

36.5.3.14 CRC Error

This error exists only for isochronous IN pipes. It sets the PINTFLAG.TRFAIL, which triggers an interrupt if PINTENCLR/SET.TRFAIL is set. The user must check the PINTSMRY to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the bank descriptor STATUS_BK for each bank and if the CRC Error bit in STATUS_BK (STATUS_BK.CRCERR) is set then the user is able to determine the origin of the CRC error. A CRC error can occur during the IN stage if the USB detects a corrupted packet. The IN packet will remain stored in the bank and PINTFLAG.TRCPT0/1 will be set.

36.5.3.15 PERR Error

This error exists for all pipes. It sets the PINTFLAG.PERR Interrupt, which triggers an interrupt if PINTFLAG.PERR is set. The user must check the PINTSMRY register to find out the pipe which can cause an interrupt.

A PERR error occurs if one of the error field in the STATUS_PIPE register in the Host pipe descriptor is set and the Error Count field in STATUS_PIPE (STATUS_PIPE.ERCNT) exceeds the maximum allowed number of Pipe error(s) as defined in Pipe Error Max Number field in CTRL_PIPE (CTRL_PIPE.PERMAX). Refer to the STATUS_PIPE register.

If one of the error field in the STATUS_PIPE register from the Host Pipe Descriptor is set and the STATUS_PIPE.ERCNT is less than the CTRL_PIPE.PERMAX, the STATUS_PIPE.ERCNT is incremented.

36.5.3.16 Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Host.

An EXTENDED LPM transaction can be transmitted by any enabled pipe. The PCFGn.PTYPE should be set to EXTENDED. Other fields as PCFG.PTOKEN, PCFG.BK and PCKSIZE.SIZE are irrelevant in this configuration. The user should also set the EXTREG.VARIABLE in the descriptor as described in Extended Register (EXTREG) register.

When the pipe is configured and enabled, an EXTENDED TOKEN followed by a LPM TOKEN are transmitted. The device responds with a valid HANDSHAKE, corrupted HANDSHAKE or no HANDSHAKE (TIME-OUT).

If the valid HANDSHAKE is an ACK, the host will immediately proceed to L1 SLEEP and the PINTFLAG.TRCT0 is set. The minimum duration of the L1 SLEEP state will be the TL1RetryAndResidency as defined in the reference document "ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum". When entering the L1 SLEEP state, the CTRLB.SOFE is cleared, avoiding Start-of-Frame generation.

If the valid HANDSHAKE is a NYET PINTFLAG.TRFAIL is set.

If the valid HANDSHAKE is a STALL the PINTFLAG.STALL is set.

If there is no HANDSHAKE or corrupted HANDSHAKE, the EXTENDED/LPM pair of TOKENS will be transmitted again until reaching the maximum number of retries as defined by the CTRL_PIPE.PERMAX in the pipe descriptor.

If the last retry returns no valid HANDSHAKE, the PINTFLAGn.PERR is set, and the STATUS_BK is updated in the pipe descriptor.

All LPM transactions, should they end up with a ACK, a NYET, a STALL or a PERR, will set the PSTATUS.PFREEZE bit, freezing the pipe before a succeeding operation. The user should unfreeze the pipe to start a new LPM transaction.

To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. See Extended Register (EXTREG) register.

When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

After resuming from the L1 SLEEP state, the bit CTRLB.SOFE is set, allowing Start-of-Frame generation.

36.6 Peripheral Dependencies

Table 36-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
USB	0x44832000	134 : EORSMDNRSN/ EORSTRST/ LPMDCONN/ LPMSUSPDDISC/ MSOF/ RAMACER/ RXSTPTXSTP0/ RXSTPTXSTP1/ RXSTPTXSTP2/ RXSTPTXSTP3/ RXSTPTXSTP4/ RXSTPTXSTP5/ RXSTPTXSTP6/ RXSTPTXSTP7/ STALLOSTALLO/ STALLOSTALL1/ STALLOSTALL2/ STALLOSTALL3/ STALLOSTALL4/ STALLOSTALL5/ STALLOSTALL6/ STALLOSTALL7/ STALL10/ STALL11/ STALL12/ STALL13/ STALL14/ STALL15/ STALL16/ STALL17/ SUSPEND/ TRFAIL0TRFAIL0/ TRFAIL0TRFAIL1/ TRFAIL0TRFAIL2/ TRFAIL0TRFAIL3/ TRFAIL0TRFAIL4/ TRFAIL0TRFAIL5/ TRFAIL0TRFAIL6/ TRFAIL0TRFAIL7/ TRFAIL1PERR0/ TRFAIL1PERR1/ TRFAIL1PERR2/ TRFAIL1PERR3/ TRFAIL1PERR4/ TRFAIL1PERR5/ TRFAIL1PERR6/ TRFAIL1PERR7/ UPRSM/ WAKEUP 135 : SOFHSOF 136 : TRCPT00/ TRCPT01/ TRCPT02/ TRCPT03/ TRCPT04/ TRCPT05/ TRCPT06/ TRCPT07 137 : TRCPT10/ TRCPT11/ TRCPT12/ TRCPT13/ TRCPT14/ TRCPT15/ TRCPT16/ TRCPT17	AHB: CLKMSK[1] MASK22 APB: CLKMSK[1] MASK23	PCHCTRL[38]	44 STATUS1 PERID12			PD_CORE_SW

36.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	7:0	MODE					RUNSTDBY	ENABLE	SWRST
0x01	Reserved									
0x02	SYNCBUSY	7:0							ENABLE	SWRST
0x03	QOSCTRL	7:0					DQOS[1:0]		CQOS[1:0]	
0x04	Reserved									
...										
0x0C										
0x0D	FSMSTATUS	7:0		FSMSTATE[6:0]						
0x0E	Reserved									
...										
0x23										
0x24	DESCADD	31:24	DESCADD[31:24]							
		23:16	DESCADD[23:16]							
		15:8	DESCADD[15:8]							
		7:0	DESCADD[7:0]							
0x28	PADCAL	15:8		TRIM[2:0]				TRANSN[4:2]		
		7:0	TRANSN[1:0]			TRANSP[4:0]				
0x2A	Reserved									
...										
0xFF										
0x0100	EPCFG - DEVICE_ENDPOINT Mode	7:0	NYETDIS	EPTYPE1[2:0]				EPTYPE0[2:0]		
0x0100	PCFG - HOST_PIPE Mode	7:0			PTYPE[2:0]			BK	PTOKEN[1:0]	
0x0101	Reserved									
...										
0x0102										
0x0103	BINTERVAL - HOST_PIPE Mode	7:0	BITINTERVAL[7:0]							
0x0104	EPSTATUSCLR - DEVICE_ENDPOINT Mode	7:0	BK1RDY	BK0RDY	STALLRQ[1:0]			CURBK	DTGLIN	DTGLOUT
0x0104	PSTATUSCLR - HOST_PIPE Mode	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0105	EPSTATUSSET - DEVICE_ENDPOINT Mode	7:0	BK1RDY	BK0RDY	STALLRQ[1:0]			CURBK	DTGLIN	DTGLOUT
0x0105	PSTATUSSET - HOST_PIPE Mode	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0106	EPSTATUS - DEVICE_ENDPOINT Mode	7:0	BK1RDY	BK0RDY	STALLRQ[1:0]			CURBK	DTGLIN	DTGLOUT
0x0106	PSTATUS - HOST_PIPE Mode	7:0	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
0x0107	EPINTFLAG - DEVICE_ENDPOINT Mode	7:0		STALL[1:0]		RXSTP	TRFAIL[1:0]		TRCPT[1:0]	
0x0107	PINTFLAG - HOST_PIPE Mode	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT[1:0]	
0x0108	EPINTENCLR - DEVICE_ENDPOINT Mode	7:0		STALL[1:0]		RXSTP	TRFAIL[1:0]		TRCPT[1:0]	
0x0108	PINTENCLR - HOST_PIPE Mode	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT[1:0]	
0x0109	EPINTENSET - DEVICE_ENDPOINT Mode	7:0		STALL[1:0]		RXSTP	TRFAIL[1:0]		TRCPT[1:0]	
0x0109	PINTENSET - HOST_PIPE Mode	7:0			STALL	TXSTP	PERR	TRFAIL	TRCPT[1:0]	

36.7.1 Control A

Name: CTRLA
Offset: 0x000
Reset: 0x00
Property: RW

Table 36-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	MODE					RUNSTDBY	ENABLE	SWRST
Access	RW					RW	RW	RW
Reset	0					0	0	0

Bit 7 – MODE Operating Mode

Value	Name	Description
0	DEVICE	Device Mode
1	HOST	Host Mode

Bit 2 – RUNSTDBY Run in Standby Mode

Bit 1 – ENABLE Enable

Bit 0 – SWRST Software Reset

36.7.2 Synchronization Busy

Name: SYNCBUSY
Offset: 0x002
Reset: 0x00
Property: R

Table 36-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable Synchronization Busy

Bit 0 – SWRST Software Reset Synchronization Busy

36.7.3 USB Quality Of Service

Name: QOSCTRL
Offset: 0x003
Reset: 0x0F
Property: RW

Table 36-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
					DQOS[1:0]		CQOS[1:0]	
Access					RW	RW	RW	RW
Reset					1	1	1	1

Bits 3:2 – DQOS[1:0] Data Quality of Service

Bits 1:0 – CQOS[1:0] Configuration Quality of Service

36.7.4 Finite State Machine Status

Name: FSMSTATUS
Offset: 0x00D
Reset: 0x01
Property: R

Table 36-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	1

Bits 6:0 – FSMSTATE[6:0] Fine State Machine Status

Value	Name	Description
0x1	OFF	OFF (L3). It corresponds to the powered-off, disconnected, and disabled state
0x2	ON	ON (L0). It corresponds to the Idle and Active states
0x4	SUSPEND	SUSPEND (L2)
0x8	SLEEP	SLEEP (L1)
0x10	DNRESUME	DNRESUME. Down Stream Resume.
0x20	UPRESUME	UPRESUME. Up Stream Resume.
0x40	RESET	RESET. USB lines Reset.

36.7.5 Descriptor Address

Name: DESCADD
Offset: 0x024
Reset: 0x00000000
Property: RW

Table 36-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DESCADD[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DESCADD[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DESCADD[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DESCADD[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DESCADD[31:0] Descriptor address Register

36.7.6 USB PAD Calibration

Name: PADCAL
Offset: 0x028
Reset: 0x0000
Property: RW

Table 36-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
		TRIM[2:0]				TRANSN[4:2]		
Access		RW	RW	RW		RW	RW	RW
Reset		0	0	0		0	0	0
Bit	7	6	5	4	3	2	1	0
	TRANSN[1:0]			TRANSP[4:0]				
Access	RW	RW						
Reset	0	0		0	0	0	0	0

Bits 14:12 – TRIM[2:0] USB Pad Trim calibration

Bits 10:6 – TRANSN[4:0] USB Pad Transn calibration

Bits 4:0 – TRANSP[4:0] USB Pad Transp calibration

36.7.7 End Point Configuration

Name: EPCFG - DEVICE_ENDPOINT Mode
Offset: 0x100
Reset: 0x00
Property: RW

Table 36-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	NYETDIS	EPTYPE1[2:0]				EPTYPE0[2:0]		
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0

Bit 7 – NYETDIS NYET Token Disable

Bits 6:4 – EPTYPE1[2:0] End Point Type1

Bits 2:0 – EPTYPE0[2:0] End Point Type0

36.7.8 End Point Configuration

Name: PCFG - HOST_PIPE Mode
Offset: 0x100
Reset: 0x00
Property: RW

Table 36-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			PTYPE[2:0]			BK	PTOKEN[1:0]	
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:3 – PTYPE[2:0] Pipe Type

Bit 2 – BK Pipe Bank

Bits 1:0 – PTOKEN[1:0] Pipe Token

36.7.9 Bus Access Period of Pipe

Name: BINTERVAL - HOST_PIPE Mode
Offset: 0x103
Reset: 0x00
Property: RW

Table 36-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BITINTERVAL[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – BITINTERVAL[7:0] Bit Interval

36.7.10 End Point Pipe Status Clear

Name: EPSTATUSCLR - DEVICE_ENDPOINT Mode
Offset: 0x104
Reset: 0x00
Property: W

Table 36-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ[1:0]			CURBK	DTGLIN	DTGLOUT
Access	W	W	W	W		W	W	W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 Ready Clear

Bit 6 – BK0RDY Bank 0 Ready Clear

Bits 5:4 – STALLRQ[1:0] Stall x Request Clear

Bit 2 – CURBK Current Bank Clear

Bit 1 – DTGLIN Data Toggle IN Clear

Bit 0 – DTGLOUT Data Toggle OUT Clear

36.7.11 End Point Pipe Status Clear

Name: PSTATUSCLR - HOST_PIPE Mode
Offset: 0x104
Reset: 0x00
Property: W

Table 36-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	W	W		W		W		R
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 Ready Clear

Bit 6 – BK0RDY Bank 0 Ready Clear

Bit 4 – PFREEZE Pipe Freeze Clear

Bit 2 – CURBK Curren Bank clear

Bit 0 – DTGL Data Toggle clear

36.7.12 End Point Pipe Status Set

Name: EPSTATUSSET - DEVICE_ENDPOINT Mode
Offset: 0x105
Reset: 0x00
Property: W

Table 36-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ[1:0]			CURBK	DTGLIN	DTGLOUT
Access	W	W	W	W		W	W	W
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 Ready Set

Bit 6 – BK0RDY Bank 0 Ready Set

Bits 5:4 – STALLRQ[1:0] Stall x Request Set

Bit 2 – CURBK Current Bank Set

Bit 1 – DTGLIN Data Toggle IN Set

Bit 0 – DTGLOUT Data Toggle OUT Set

36.7.13 End Point Pipe Status Set

Name: PSTATUSSET - HOST_PIPE Mode
Offset: 0x105
Reset: 0x00
Property: W

Table 36-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	W	W		W		W		W
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 Ready Set

Bit 6 – BK0RDY Bank 0 Ready Set

Bit 4 – PFREEZE Pipe Freeze Set

Bit 2 – CURBK Current Bank Set

Bit 0 – DTGL Data Toggle Set

36.7.14 End Point Pipe Status

Name: EPSTATUS - DEVICE_ENDPOINT Mode
Offset: 0x106
Reset: 0x00
Property: R

Table 36-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY	STALLRQ[1:0]			CURBK	DTGLIN	DTGLOUT
Access	R	R	R	R		R	R	R
Reset	0	0	0	0		0	0	0

Bit 7 – BK1RDY Bank 1 ready

Bit 6 – BK0RDY Bank 0 ready

Bits 5:4 – STALLRQ[1:0] Stall x Request

Bit 2 – CURBK Current Bank

Bit 1 – DTGLIN Data Toggle In

Bit 0 – DTGLOUT Data Toggle Out

36.7.15 End Point Pipe Status

Name: PSTATUS - HOST_PIPE Mode
Offset: 0x106
Reset: 0x00
Property: R

Table 36-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	BK1RDY	BK0RDY		PFREEZE		CURBK		DTGL
Access	R	R		R		R		R
Reset	0	0		0		0		0

Bit 7 – BK1RDY Bank 1 ready

Bit 6 – BK0RDY Bank 0 ready

Bit 4 – PFREEZE Pipe Freeze

Bit 2 – CURBK Current Bank

Bit 0 – DTGL Data Toggle

36.7.16 End Point Interrupt Flag

Name: EPINTFLAG - DEVICE_ENDPOINT Mode
Offset: 0x107
Reset: 0x00
Property: RW

Table 36-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
		STALL[1:0]		RXSTP	TRFAIL[1:0]		TRCPT[1:0]	
Access				RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 6:5 – STALL[1:0] Stall x In/out

Bit 4 – RXSTP Received Setup

Bits 3:2 – TRFAIL[1:0] Error Flow x

Bits 1:0 – TRCPT[1:0] Transfer Complete x

36.7.17 Pipe Interrupt Flag

Name: PINTFLAG - HOST_PIPE Mode
Offset: 0x107
Reset: 0x00
Property: RW

Table 36-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT[1:0]	
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit 5 – STALL Stall Interrupt Flag

Bit 4 – TXSTP Transmit Setup Interrupt Flag

Bit 3 – PERR Pipe Error Interrupt Flag

Bit 2 – TRFAIL Error Flow Interrupt Flag

Bits 1:0 – TRCPT[1:0] Transfer Complete x Interrupt Flag

36.7.18 End Point Interrupt Clear Flag

Name: EPINTENCLR - DEVICE_ENDPOINT Mode
Offset: 0x108
Reset: 0x00
Property: RW

Table 36-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
		STALL[1:0]		RXSTP	TRFAIL[1:0]		TRCPT[1:0]	
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 6:5 – STALL[1:0] Stall x In/Out Interrupt Disable

Bit 4 – RXSTP Received Setup Interrupt Disable

Bits 3:2 – TRFAIL[1:0] Error Flow x Interrupt Disable

Bits 1:0 – TRCPT[1:0] Transfer Complete x Interrupt Disable

36.7.19 Pipe Interrupt Flag Clear

Name: PINTENCLR - HOST_PIPE Mode
Offset: 0x108
Reset: 0x00
Property: RW

Table 36-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT[1:0]	
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit 5 – STALL Stall Interrupt Disable

Bit 4 – TXSTP Transmit Setup Interrupt Disable

Bit 3 – PERR Pipe Error Interrupt Disable

Bit 2 – TRFAIL Error Flow Interrupt Disable

Bits 1:0 – TRCPT[1:0] Transfer Complete x Disable

36.7.20 End Point Interrupt Set Flag

Name: EPINTENSET - DEVICE_ENDPOINT Mode
Offset: 0x109
Reset: 0x00
Property: RW

Table 36-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
		STALL[1:0]		RXSTP	TRFAIL[1:0]		TRCPT[1:0]	
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 6:5 – STALL[1:0] Stall x In/out Interrupt enable

Bit 4 – RXSTP Received Setup Interrupt Enable

Bits 3:2 – TRFAIL[1:0] Error Flow x Interrupt Enable

Bits 1:0 – TRCPT[1:0] Transfer Complete x Interrupt Enable

36.7.21 Pipe Interrupt Flag Set

Name: PINTENSET - HOST_PIPE Mode
Offset: 0x109
Reset: 0x00
Property: RW

Table 36-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
			STALL	TXSTP	PERR	TRFAIL	TRCPT[1:0]	
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bit 5 – STALL Stall Interrupt Enable

Bit 4 – TXSTP Transmit Setup Interrupt Enable

Bit 3 – PERR Pipe Error Interrupt Enable

Bit 2 – TRFAIL Error Flow Interrupt Enable

Bits 1:0 – TRCPT[1:0] Transfer Complete x Interrupt Enable

36.8 Register Summary - DEVICE

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x07	Reserved									
0x08	CTRLB	15:8					LPMHDSK[1:0]		GNAK	OPMODE2
		7:0	TSTPCKT	TSTK	TSTJ	NREPLY	SPDCONF[1:0]		UPRSM	DETACH
0x0A	DADD	7:0	ADDEN				DADD[6:0]			
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0D ... 0x0F	Reserved									
0x10	FNUM	15:8	FNCERR				FNUM[10:5]			
		7:0			FNUM[4:0]				MFNUM[2:0]	
0x12 ... 0x13	Reserved									
0x14	INTENCLR	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
0x16 ... 0x17	Reserved									
0x18	INTENSET	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
0x1A ... 0x1B	Reserved									
0x1C	INTFLAG	15:8							LPMSUSP	LPMNYET
		7:0	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
0x1E ... 0x1F	Reserved									
0x20	EPINTSMRY	15:8					EPINT[15:8]			
		7:0					EPINT[7:0]			

36.8.1 Control B

Name: CTRLB
Offset: 0x008
Reset: 0x0001
Property: RW

Bit	15	14	13	12	11	10	9	8
					LPMHDSK[1:0]		GNAK	OPMODE2
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	TSTPCKT	TSTK	TSTJ	NREPLY	SPDCONF[1:0]		UPRSM	DETACH
Access	RW	RW	RW		RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	1

Bits 11:10 – LPMHDSK[1:0] Link Power Management Handshake

These bits select the Link Power Management Handshake configuration.

Value	Name	Description
0x0	NO	No handshake. LPM is not supported.
0x1	ACK	ACK
0x2	NYET	NYET
0x3	-	Reserved

Value	Name	Description
0	NO	No handshake. LPM is not supported
1	ACK	ACK
2	NYET	NYET
3	STALL	STALL

Bit 9 – GNAK Global NAK

This bit configures the operating mode of the NAK.
This bit is not synchronized.

Value	Description
0	The handshake packet reports the status of the USB transaction
1	A NAK handshake is answered for each USB transaction regardless of the current endpoint memory bank status

Bit 8 – OPMODE2 Specific Operational Mode

Bit 7 – TSTPCKT Test packet mode

Bit 6 – TSTK Test mode K

Bit 5 – TSTJ Test mode J

Bit 4 – NREPLY No Reply

This bit is cleared by hardware when receiving a SETUP packet.
This bit has no effect for any other endpoint but endpoint 0.

Value	Description
0	Disable the "NO_REPLY" feature; Any transaction to endpoint 0 will be handled according to the USB2.0 standard.
1	Enable the "NO_REPLY" feature; Any transaction to endpoint 0 will be ignored except SETUP.

Bits 3:2 – SPDCONF[1:0] Speed Configuration
These bits select the speed configuration.

Value	Name	Description
0x0	FS	FS : Full Speed
0x1	LS	LS : Low Speed
0x2	HS	HS : High Speed capable
0x3	HSTM	HSTM: High Speed Test Mode (force high-speed mode for test mode)

Bit 1 – UPRSM Upstream Resume
This bit is cleared when the USB receives a USB reset or once the upstream resume has been sent.

Value	Description
0	Writing a zero to this bit has no effect.
1	Writing a one to this bit will generate an upstream resume to the host for a remote wakeup.

Bit 0 – DETACH Detach

Value	Description
0	The device is attached to the USB bus so that communications may occur.
1	It is the default value at reset. The internal device pull-ups are disabled, removing the device from the USB bus.

36.8.2 Device Address

Name: DADD
Offset: 0x00A
Reset: 0x00
Property: RW

Bit	7	6	5	4	3	2	1	0
	ADDEN	DADD[6:0]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 7 – ADDEN Device Address Enable

This bit is cleared when a USB reset is received.

Value	Description
0	Writing a zero will deactivate the DADD field (USB device address) and return the device to default address 0.
1	Writing a one will activate the DADD field (USB device address).

Bits 6:0 – DADD[6:0] Device Address

These bits define the device address. The DADD register is reset when a USB reset is received.

36.8.3 Status

Name: STATUS
Offset: 0x00C
Reset: 0x40
Property: R

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			R	R		
Reset	0	1			0	0		

Bits 7:6 – LINESTATE[1:0] USB Line State Status

Value	Name	Description
0x0	0	SE0/RESET
0x1	1	FS-J or LS-K State
0x2	2	FS-K or LS-J State

Bits 3:2 – SPEED[1:0] Speed Status

Value	Name	Description
0x0	FS	Full-speed mode
0x1	LS	Low-speed mode
0x2	HS	High-speed mode

36.8.4 Device Frame Number

Name: FNUM
Offset: 0x010
Reset: 0x0000
Property: R

Bit	15	14	13	12	11	10	9	8
	FNCERR			FNUM[10:5]				
Access	R		R	R	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]					MFNUM[2:0]		
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 15 – FNCERR Frame Number CRC Error

This bit is cleared upon receiving a USB reset.

This bit is set when a corrupted frame number (or micro-frame number) is received.

This bit and the SOF interrupt bit are updated at the same time.

Bits 13:3 – FNUM[10:0] Frame Number

These bits are cleared upon receiving a USB reset.

These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.

Bits 2:0 – MFNUM[2:0] Micro Frame Number

36.8.5 Device Interrupt Enable Clear

Name: INTENCLR
Offset: 0x014
Reset: 0x0000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Link Power Management Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled and an interrupt request will be generated when the Link Power Management Suspend interrupt Flag is set.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Link Power Management Not Yet interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled and an interrupt request will be generated when the Link Power Management Not Yet interrupt Flag is set.

Bit 7 – RAMACER Ram Access Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set.

Bit 6 – UPRSM Upstream Resume Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.

Value	Description
1	The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set.

Bit 5 – EORSM End Of Resume Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the End Of Resume interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled and an interrupt request will be generated when the End Of Resume interrupt Flag is set.

Bit 4 – WAKEUP Wake Up Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set.

Bit 3 – EORST End of Reset Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the End of Reset interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled and an interrupt request will be generated when the End of Reset interrupt Flag is set.

Bit 2 – SOF Start Of Frame Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Start-of-Frame interrupt Flag is set.

Bit 1 – MSOF Micro Start of Frame Interrupt Enable in High Speed Mode

Bit 0 – SUSPEND Suspend Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Suspend Interrupt Enable bit and disable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled and an interrupt request will be generated when the Suspend interrupt Flag is set.

36.8.6 Device Interrupt Enable Set

Name: INTENSET
Offset: 0x018
Reset: 0x0000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 9 – LPMSUSP Link Power Management Suspend Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Link Power Management Suspend Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Suspend interrupt is disabled.
1	The Link Power Management Suspend interrupt is enabled.

Bit 8 – LPMNYET Link Power Management Not Yet Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Link Power Management Not Yet interrupt bit and enable the corresponding interrupt request.

Value	Description
0	The Link Power Management Not Yet interrupt is disabled.
1	The Link Power Management Not Yet interrupt is enabled.

Bit 7 – RAMACER Ram Access Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the RAM Access Enable bit and enable the corresponding interrupt request.

Value	Description
0	The RAM Access interrupt is disabled.
1	The RAM Access interrupt is enabled.

Bit 6 – UPRSM Upstream Resume Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Upstream Resume Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Upstream Resume interrupt is disabled.
1	The Upstream Resume interrupt is enabled.

Bit 5 – EORSM End Of Resume Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End Of Resume interrupt is disabled.
1	The End Of Resume interrupt is enabled.

Bit 4 – WAKEUP Wake Up Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Wake Up interrupt is disabled.
1	The Wake Up interrupt is enabled.

Bit 3 – EORST End of Reset Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The End of Reset interrupt is disabled.
1	The End of Reset interrupt is enabled.

Bit 2 – SOF Start Of Frame Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Start-of-Frame interrupt is disabled.
1	The Start-of-Frame interrupt is enabled.

Bit 1 – MSOF Micro Start of Frame Interrupt Enable in High Speed Mode

Bit 0 – SUSPEND Suspend Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

Value	Description
0	The Suspend interrupt is disabled.
1	The Suspend interrupt is enabled.

36.8.7 Device Interrupt Flag

Name: INTFLAG
Offset: 0x01C
Reset: 0x0000
Property: RW

Bit	15	14	13	12	11	10	9	8
							LPMSUSP	LPMNYET
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSPEND
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 9 – LPMSUSP Link Power Management Suspend

This flag is cleared by writing a one to the flag.

This flag is set when the USB module acknowledge a Link Power Management Transaction (ACK handshake) and has entered the Suspended state and will generate an interrupt if INTENSET.LPMSUSP is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the LPMSUSP Interrupt Flag.

Bit 8 – LPMNYET Link Power Management Not Yet

This flag is cleared by writing a one to the flag.

This flag is set when the USB module acknowledges a Link Power Management Transaction (handshake is NYET) and will generate an interrupt if INTENSET.LPMNYET is one.

Writing a zero to this bit has no effect.

Writing a one to this bit clears the LPMNYET Interrupt Flag.

Bit 7 – RAMACER Ram Access

This flag is cleared by writing a one to the flag.

This flag is set when a RAM access underflow error occurs during IN data stage. This bit will generate an interrupt if INTENSET.RAMACER is one.

Writing a zero to this bit has no effect.

Bit 6 – UPRSM Upstream Resume

This flag is cleared by writing a one to the flag.

This flag is set when the USB sends a resume signal called “Upstream Resume” and will generate an interrupt if INTENSET.UPRSM is one.

Writing a zero to this bit has no effect.

Bit 5 – EORSM End Of Resume

This flag is cleared by writing a one to the flag.

This flag is set when the USB detects a valid “End of Resume” signal initiated by the host and will generate an interrupt if INTENSET.EORSM is one.

Writing a zero to this bit has no effect.

Bit 4 – WAKEUP Wake Up

This flag is cleared by writing a one to the flag.

This flag is set when the USB is reactivated by a filtered non-idle signal from the lines and will generate an interrupt if INTENSET.WAKEUP is one.

Writing a zero to this bit has no effect.

Bit 3 – EORST End of Reset

This flag is cleared by writing a one to the flag.

This flag is set when a USB “End of Reset” has been detected and will generate an interrupt if INTENSET.EORST is one.

Writing a zero to this bit has no effect.

Bit 2 – SOF Start Of Frame

This flag is cleared by writing a one to the flag.

This flag is set when a USB “Start-of-Frame” has been detected (every 1 ms) and will generate an interrupt if INTENSET.SOF is one.

The FNUM is updated.

Writing a zero to this bit has no effect.

Bit 1 – MSOF Micro Start of Frame in High Speed Mode

Bit 0 – SUSPEND Suspend

This flag is cleared by writing a one to the flag.

This flag is set when a USB “Suspend” idle state has been detected for 3 frame periods (J state for 3 ms) and will generate an interrupt if INTENSET.SUSPEND is one.

Writing a zero to this bit has no effect.

36.8.8 End Point Interrupt Summary

Name: EPINTSMRY
Offset: 0x020
Reset: 0x0000
Property: R

Bit	15	14	13	12	11	10	9	8
	EPINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EPINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EPINT[15:0] End Point x Interrupt

The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n.
This bit will be cleared when no interrupts are pending for EndPoint n.

36.9 Register Summary - HOST

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00 ... 0x07	Reserved									
0x08	CTRLB	15:8					L1RESUME	VBUSOK	BUSRESET	SOFE
		7:0		TSTK	TSTJ	AUTORESUME	SPDCONF[1:0]		RESUME	
0x0A	HSOFC	7:0	FLENCE				FLENC[3:0]			
0x0B	Reserved									
0x0C	STATUS	7:0	LINESTATE[1:0]				SPEED[1:0]			
0x0D ... 0x0F	Reserved									
0x10	FNUM	15:8			FNUM[10:5]					
		7:0	FNUM[4:0]					MFNUM[2:0]		
0x12	FLENHIGH	7:0	FLENHIGH[7:0]							
0x13	Reserved									
0x14	INTENCLR	15:8							DDISC	DCONN
		7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x16 ... 0x17	Reserved									
0x18	INTENSET	15:8							DDISC	DCONN
		7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x1A ... 0x1B	Reserved									
0x1C	INTFLAG	15:8							DDISC	DCONN
		7:0	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
0x1E ... 0x1F	Reserved									
0x20	PINTSMRY	15:8	EPINT[15:8]							
		7:0	EPINT[7:0]							

36.9.1 Control B

Name: CTRLB
Offset: 0x008
Reset: 0x0000
Property: RW

Bit	15	14	13	12	11	10	9	8
					L1RESUME	VBUSOK	BUSRESET	SOFE
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
		TSTK	TSTJ	AUTORESUME	SPDCONF[1:0]		RESUME	
Access		RW	RW	RW	RW	RW	RW	
Reset		0	0	0	0	0	0	

Bit 11 – L1RESUME Send L1 Resume

Writing 0 to this bit has no effect.

1: Generates a USB L1 Resume on the USB bus. This bit should only be set when the Start-of-Frame generation is enabled (SOFE bit set). The duration of the USB L1 Resume is defined by the EXTREG.VARIABLE[7:4] bits field also known as BESL (See LPM ECN).

This bit is cleared when the USB L1 Resume has been sent or when a USB reset is requested.

Bit 10 – VBUSOK VBUS is OK

This notifies the USB HOST that USB operations can be started. When this bit is zero and even if the USB HOST is configured and enabled, HOST operation is halted. Setting this bit will allow HOST operation when the USB is configured and enabled.

Value	Description
0	The USB module is notified that the VBUS on the USB line is not powered.
1	The USB module is notified that the VBUS on the USB line is powered.

Bit 9 – BUSRESET Send USB Reset

Value	Description
0	Reset generation is disabled. It is written to zero when the USB reset is completed or when a device disconnection is detected. Writing zero has no effect.
1	Generates a USB Reset on the USB bus.

Bit 8 – SOFE Start of Frame Generation Enable

Value	Description
0	The SOF generation is disabled and the USB bus is in suspend state.
1	Generates SOF on the USB bus in full speed and keep it alive in low speed mode. This bit is automatically set at the end of a USB reset (INTFLAG.RST) or at the end of a downstream resume (INTFLAG.DNRSM) or at the end of L1 resume.

Bit 6 – TSTK Test mode K

Bit 5 – TSTJ Test mode J

Bit 4 – AUTORESUME Auto Resume Enable

Value	Description
0	The Auto Resume is disabled.
1	Enable Auto Resume

Bits 3:2 – SPDCONF[1:0] Speed Configuration for Host

These bits select the host speed configuration as shown below

Value	Name	Description
0x0	NORMAL	Normal mode: the host starts in full-speed mode and performs a high-speed reset to switch to the high speed mode if the downstream peripheral is high-speed capable.
0x3	FS	Full-speed: the host remains in full-speed mode whatever is the peripheral speed capability. Relevant in UTMI mode only.

Bit 1 – RESUME Send USB Resume

Writing 0 to this bit has no effect.

1: Generates a USB Resume on the USB bus.

This bit is cleared when the USB Resume has been sent or when a USB reset is requested.

36.9.2 Host Start Of Frame Control

Name: HSOFC
Offset: 0x00A
Reset: 0x00
Property: RW

During a very short period just before transmitting a Start-of-Frame, this register is locked. Thus, after writing, it is recommended to check the register value, and write this register again if necessary. This register is cleared upon a USB reset.

Bit	7	6	5	4	3	2	1	0
	FLENCE				FLENC[3:0]			
Access	RW				RW	RW	RW	RW
Reset	0				0	0	0	0

Bit 7 – FLENCE Frame Length Control Enable

When this bit is '1', the time between Start-of-Frames can be tuned by up to +/-0.06% using FLENC[3:0].

Note: In Low Speed mode, FLENCE must be '0'.

Value	Description
0	Start-of-Frame is generated every 1ms.
1	Start-of-Frame generation depends on the signed value of FLENC[3:0]. USB Start-of-Frame period equals 1ms + (FLENC[3:0]/12000)ms

Bits 3:0 – FLENC[3:0] Frame Length Control

These bits define the signed value of the 4-bit FLENC that is added to the Internal Frame Length when FLENCE is '1'. The internal Frame length is the top value of the frame counter when FLENCE is zero.

36.9.3 Status

Name: STATUS
Offset: 0x00C
Reset: 0x00
Property: RW

Bit	7	6	5	4	3	2	1	0
	LINESTATE[1:0]				SPEED[1:0]			
Access	R	R			RW	RW		
Reset	0	0			0	0		

Bits 7:6 – LINESTATE[1:0] USB Line State Status

Bits 3:2 – SPEED[1:0] Speed Status

36.9.4 Host Frame Number

Name: FNUM
Offset: 0x010
Reset: 0x0000
Property: RW

Bit	15	14	13	12	11	10	9	8
			FNUM[10:5]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]					MFNUM[2:0]		
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 13:3 – FNUM[10:0] Frame Number

These bits contains the current SOF number.

These bits can be written by software to initialize a new frame number value. In this case, at the next SOF, the FNUM field takes its new value.

As the FNUM register lies across two consecutive byte addresses, writing byte-wise (8-bits) to the FNUM register may produce incorrect frame number generation. It is recommended to write FNUM register word-wise (32-bits) or half-word-wise (16-bits).

Bits 2:0 – MFNUM[2:0] Micro Frame Number

36.9.5 Host Frame Length

Name: FLENHIGH
Offset: 0x012
Reset: 0x00
Property: R

Bit	7	6	5	4	3	2	1	0
	FLENHIGH[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – FLENHIGH[7:0] Frame Length

These bits contains the 8 high-order bits of the internal frame counter.

Table 36-23. Counter Description vs. Speed

Host Register STATUS.SPEED	Description
Full Speed	With a USB clock running at 12MHz, counter length is 12000 to ensure a SOF generation every 1 ms.

36.9.6 Host Interrupt Enable Clear

Name: INTENCLR
Offset: 0x014
Reset: 0x0000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENSET) register.

This register is cleared by USB reset or when PEN[n] is zero.

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Disable

Bit 8 – DCONN Device Connection Interrupt Disable

Bit 7 – RAMACER Ram Access Interrupt Disable

Bit 6 – UPRSM Upstream Resume from Device Interrupt Disable

Bit 5 – DNRSM DownStream to Device Interrupt Disable

Bit 4 – WAKEUP Wake Up Interrupt Disable

Bit 3 – RST BUS Reset Interrupt Disable

Bit 2 – HSOF Host Start Of Frame Interrupt Disable

36.9.7 Host Interrupt Enable Set

Name: INTENSET
Offset: 0x018
Reset: 0x0000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENCLR) register.

This register is cleared by USB reset or when PEN[n] is zero.

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection Interrupt Enable

Bit 8 – DCONN Link Power Management Interrupt Enable

Bit 7 – RAMACER Ram Access Interrupt Enable

Bit 6 – UPRSM Upstream Resume from the device Interrupt Enable

Bit 5 – DNRSM DownStream to the Device Interrupt Enable

Bit 4 – WAKEUP Wake Up Interrupt Enable

Bit 3 – RST Bus Reset Interrupt Enable

Bit 2 – HSOF Host Start Of Frame Interrupt Enable

36.9.8 Host Interrupt Flag

Name: INTFLAG
Offset: 0x01C
Reset: 0x0000
Property: RW

Bit	15	14	13	12	11	10	9	8
							DDISC	DCONN
Access							RW	RW
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	RAMACER	UPRSM	DNRSM	WAKEUP	RST	HSOF		
Access	RW	RW	RW	RW	RW	RW		
Reset	0	0	0	0	0	0		

Bit 9 – DDISC Device Disconnection

Bit 8 – DCONN Device Connection

Bit 7 – RAMACER Ram Access

Bit 6 – UPRSM Upstream Resume from the Device

Bit 5 – DNRSM Downstream

Bit 4 – WAKEUP Wake Up

Bit 3 – RST Bus Reset

Bit 2 – HSOF Host Start Of Frame

36.9.9 Pipe Interrupt Summary

Name: PINTSMRY
Offset: 0x020
Reset: 0x0000
Property: R

Bit	15	14	13	12	11	10	9	8
	EPINT[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EPINT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – EPINT[15:0] Pipe x Interrupt

36.10 USB Descriptor Register Summary

36.10.1 Endpoint Bank, Address of Data Buffer

Name: ADDR - DEVICE_DESC_BANK Mode
Offset: 0x000
Property: RW

Table 36-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								

Bits 31:0 – ADDR[31:0] Address of data buffer

36.10.2 Host Bank, Address of Data Buffer

Name: ADDR - HOST_DESC_BANK Mode
Offset: 0x000
Property: RW

Table 36-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	ADDR[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	23	22	21	20	19	18	17	16
	ADDR[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	15	14	13	12	11	10	9	8
	ADDR[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	7	6	5	4	3	2	1	0
	ADDR[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								

Bits 31:0 – ADDR[31:0] Address of data buffer

36.10.3 Endpoint Bank, Packet Size

Name: PCKSIZE - DEVICE_DESC_BANK Mode
Offset: 0x004
Property: RW

Table 36-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	7	6	5	4	3	2	1	0
	BYTE_COUNT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								

Bit 31 – AUTO_ZLP Automatic Zero Length Packet

Bits 30:28 – SIZE[2:0] Endpoint size

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multi Packet In or Out size

Bits 13:0 – BYTE_COUNT[13:0] Byte Count

36.10.4 Host Bank, Packet Size

Name: PCKSIZE - HOST_DESC_BANK Mode
Offset: 0x004
Property: RW

Table 36-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	AUTO_ZLP	SIZE[2:0]			MULTI_PACKET_SIZE[13:10]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	23	22	21	20	19	18	17	16
	MULTI_PACKET_SIZE[9:2]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	15	14	13	12	11	10	9	8
	MULTI_PACKET_SIZE[1:0]		BYTE_COUNT[13:8]					
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								
Bit	7	6	5	4	3	2	1	0
	BYTE_COUNT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								

Bit 31 – AUTO_ZLP Automatic Zero Length Packet

Bits 30:28 – SIZE[2:0] Pipe size

Bits 27:14 – MULTI_PACKET_SIZE[13:0] Multi Packet In or Out size

Bits 13:0 – BYTE_COUNT[13:0] Byte Count

36.10.5 Endpoint Bank, Extended

Name: EXTREG - DEVICE_DESC_BANK Mode
Offset: 0x008
Property: RW

Table 36-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 14:4 – VARIABLE[10:0] Variable field send with extended token

Bits 3:0 – SUBPID[3:0] SUBPID field send with extended token

36.10.6 Host Bank, Extended

Name: EXTREG - HOST_DESC_BANK Mode
Offset: 0x008
Property: RW

Table 36-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 14:4 – VARIABLE[10:0] Variable field send with extended token

Bits 3:0 – SUBPID[3:0] SUBPID field send with extended token

36.10.7 Endpoint Bank, Status of Bank

Name: STATUS_BK - DEVICE_DESC_BANK Mode
Offset: 0x00A
Property: RW

Table 36-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							RW	RW
Reset								

Bit 1 – ERRORFLOW Error Flow Status

Bit 0 – CRCERR CRC Error Status

36.10.8 Host Bank, Status of Bank

Name: STATUS_BK - HOST_DESC_BANK Mode
Offset: 0x00A
Property: RW

Table 36-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
							ERRORFLOW	CRCERR
Access							RW	
Reset								

Bit 1 – ERRORFLOW Error Flow Status

Bit 0 – CRCERR CRC Error Status

36.10.9 Host Bank, Host Control Pipe

Name: CTRL_PIPE - HOST_DESC_BANK Mode
Offset: 0x00C
Reset: 0x0000
Property: RW

Table 36-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PERMAX[3:0]				PEPNUM[3:0]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
		PDADDR[6:0]						
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0

Bits 15:12 – PERMAX[3:0] Pipe Error Max Number

Bits 11:8 – PEPNUM[3:0] Pipe Endpoint Number

Bits 6:0 – PDADDR[6:0] Pipe Device Address

36.10.10 Host Bank, Host Status Pipe

Name: STATUS_PIPE - HOST_DESC_BANK Mode
Offset: 0x00E
Property: RW

Table 36-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ERCNT[2:0]			CRC16ER	TOUTER	PIDER	DAPIDER	DTGLER
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset								

Bits 7:5 – ERCNT[2:0] Pipe Error Count

Bit 4 – CRC16ER CRC16 Error

Bit 3 – TOUTER Time Out Error

Bit 2 – PIDER PID Error

Bit 1 – DAPIDER Data PID Error

Bit 0 – DTGLER Data Toggle Error

37. Configurable Custom Logic (CCL)

37.1 Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each LookUp Table (LUT) consists of three inputs, a truth table, an optional synchronizer/filter, and an optional edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1, LUT2/LUT3 etc.) outputs, enabling complex waveform generation.

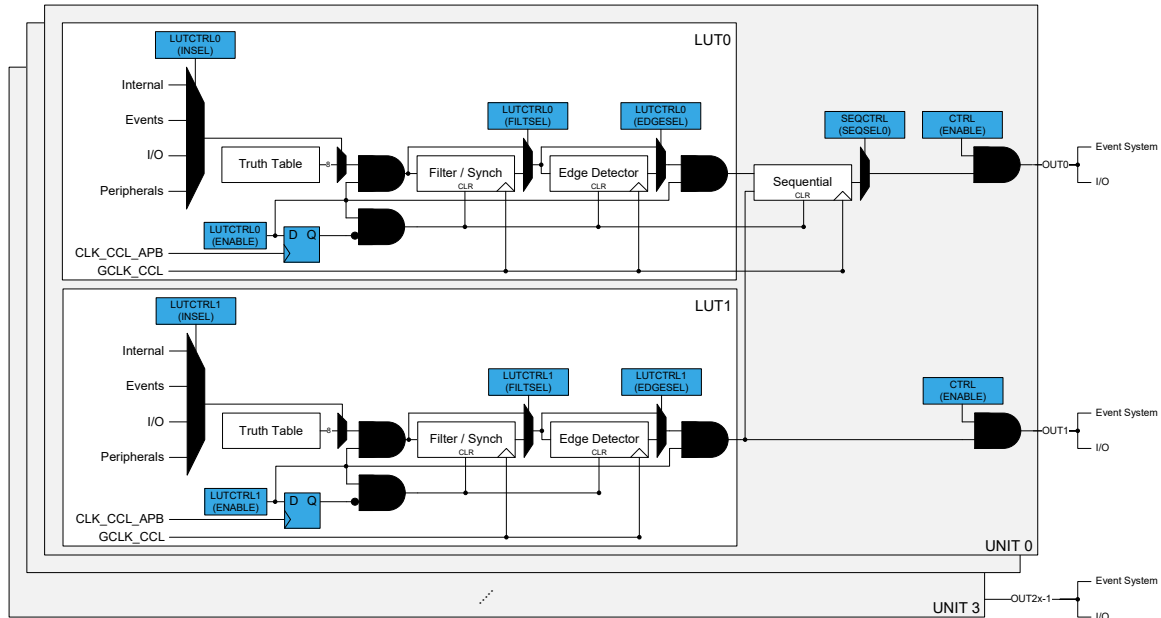
37.2 Features

The following are key features of the CCL module:

- Glue logic for general purpose PCB design
- Four programmable LookUp Tables (LUTs)
- Combinatorial logic functions:
AND, NAND, OR, NOR, XOR, XNOR, NOT
- Sequential logic functions:
Gated D Flip-Flop, JK Flip-Flop, gated D Latch, RS Latch
- Flexible LUT inputs selection:
 - I/Os
 - Events
 - Internal peripherals
 - Subsequent LUT output
- Output can be connected to the I/O pins or the Event System
- Optional synchronizer, filter, or edge detector available on each LUT output

37.3 Block Diagram

Figure 37-1. Configurable Custom Logic



37.4 Signal Description

Pin Name	Type	Description
OUT[3:0]	Digital output	Output from lookup table
IN[11:0]	Digital input	Input to lookup table

For additional information on the pin mapping for this peripheral, refer to “I/O Multiplexing and Considerations”. One signal can be mapped on several pins.

37.5 I/O Lines

The CCL can take inputs and generate output through I/O pins. For this to function properly, the I/O pins must be configured to be used by a Look Up Table (LUT).

37.6 Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

37.7 Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the Main Clock module, MCLK (refer to the Main Clock (MCLK)), and the default state of CLK_CCL_APB can be found in Peripheral Clock Masking .

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using input events, filter, edge detection or sequential logic. GCLK_CCL is required when input events, a filter, an edge detector, or a sequential sub-module is enabled. Refer to the Generic Clock Controller (GCLK) for additional information.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

37.8 Events

The CCL can use events from other peripherals and generate events that can be used by other peripherals. For this feature to function, the events have to be configured properly. Refer to the Chapter Event System (EVSYS) for additional information about the event users and event generators.

37.9 Debug Operation

When the CPU is halted in Debug mode the CCL continues normal operation. However, the CCL cannot be halted when the CPU is halted in Debug mode. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

37.10 Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC). Refer to the Chapter Peripheral Access Controller (PAC) for additional information.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

37.11 Functional Description

37.11.1 Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constraints by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

37.11.2 Operation

37.11.2.1 Initialization

The following bits are enable-protected, meaning that they can only be written when both the CCL and the corresponding even LUT are disabled (CTRL.ENABLE=0 and LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when both the CCL and the corresponding even LUT are disabled (CTRL.ENABLE=0 and LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

37.11.2.2 Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to Control (CTRL) register for details.

Note: When the Peripheral Access Controller (PAC) is enabled for the CCL module (WRCTRL.PERID) and a Software Reset (CTRL.SWRST) is executed, a PAC protection error will be generated. If the PAC interrupt is enabled, the interrupt flag for the CCL module (INTFLAG.CCL) should be cleared.

37.11.2.3 Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in the following figure. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

Figure 37-2. Truth Table Output Value Selection

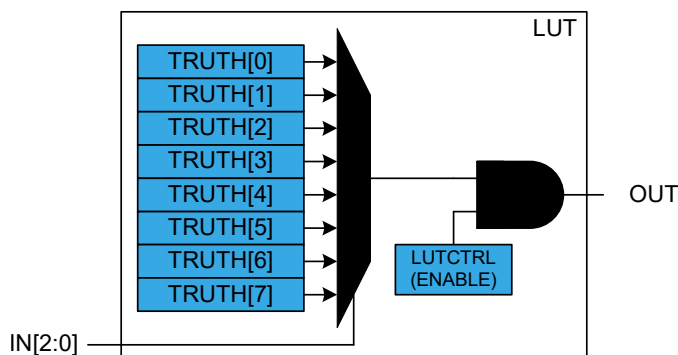


Table 37-1. Truth Table of LUT

IN[2]	IN[1]	IN[0]	OUT
0	0	0	TRUTH[0]
0	0	1	TRUTH[1]
0	1	0	TRUTH[2]
0	1	1	TRUTH[3]
1	0	0	TRUTH[4]
1	0	1	TRUTH[5]
1	1	0	TRUTH[6]
1	1	1	TRUTH[7]

37.11.2.4 Truth Table Inputs Selection

Input Overview

The inputs can be individually:

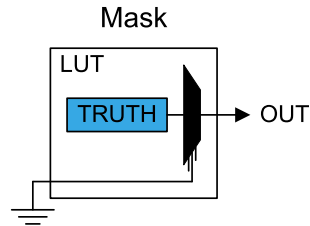
- Masked
- Driven by peripherals:
 - Analog comparator output (AC)
 - Timer/Counters waveform outputs (TC)
 - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input 'y' of LUT x is configured by writing the Input 'y' Source Selection bit in the LUT x Control register (LUTCTRLx.INSELY).

Masked Inputs (MASK)

When a LUT input is masked (LUTCTRLx.INSELY = MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

Figure 37-3. Masked Input Selection



Internal Feedback Inputs (FEEDBACK)

When selected ($LUTCTRLx.INSELY = FEEDBACK$), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

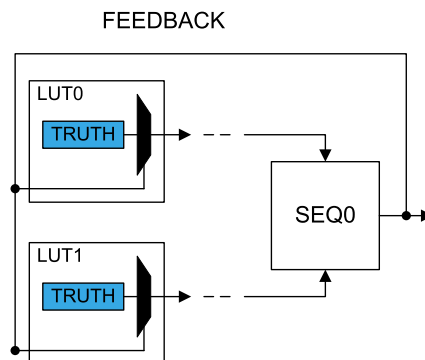
$$IN[2N][i] = SEQ[N]$$

$$IN[2N+1][i] = SEQ[N]$$

With N representing the sequencer number and $i=0,1,2$ representing the LUT input index.

For additional information, refer to Sequential Logic.

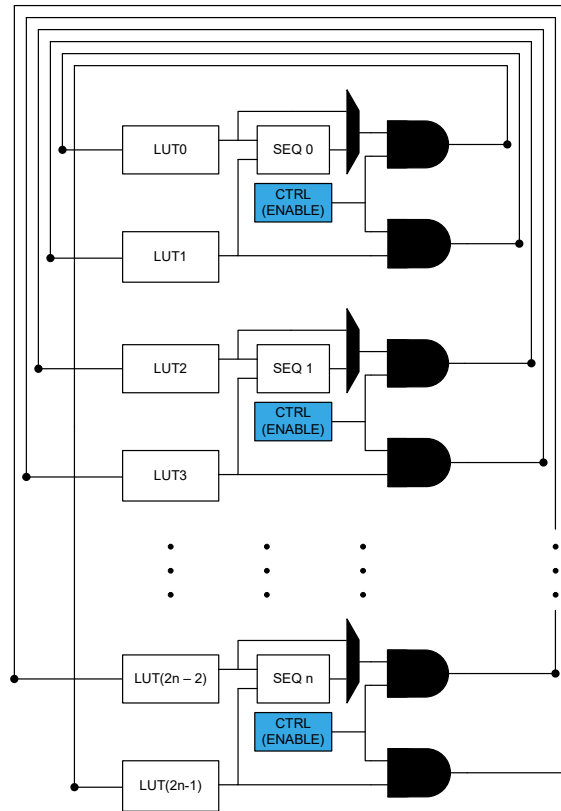
Figure 37-4. Feedback Input Selection



Linked LUT (LINK)

When selected ($LUTCTRLx.INSELY=LINK$), the subsequent LUT output is used as the LUT input (for example, LUT2 is the input for LUT1), as shown in the figure below:

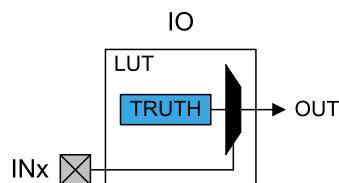
Figure 37-5. Linked LUT Input Selection



I/O Pin Inputs (IO)

When the I/O pin is selected as LUT input (LUTCTRLx.INSELY = IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

Figure 37-6. I/O Pin Input Selection



Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELY=AC).

The analog comparator outputs are distributed following the formula:

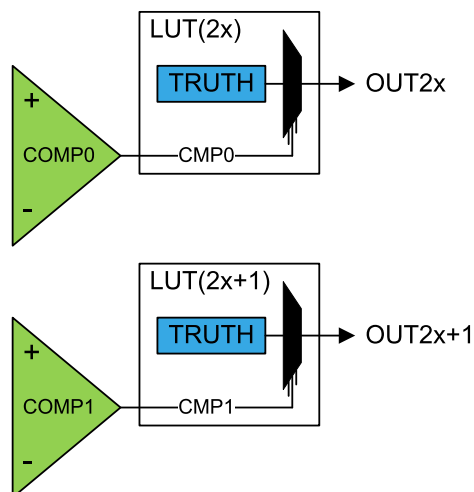
$$IN[N][i] = AC[N \% \text{ComparatorOutput\_Number}]$$

With N representing the LUT number and $i=[0,1,2]$ representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below.

Figure 37-7. AC Input Selection



Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (that is, IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

Note:

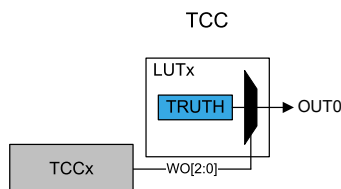
The TCC selection for each LUT follows the formula:

$$IN[N][i] = TCC[N \% TCC\_Instance\_Number].WO[i]$$

Where N represents the LUT number and i represents the LUT input index ($i=0,1,2$).

Before selecting the waveform outputs, the TCC must be configured first.

Figure 37-8. TCC Input Selection



Serial Communication Output Transmit Inputs (SERCOM)

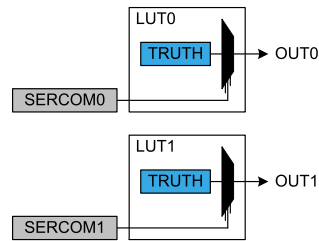
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$$IN[N][i] = SERCOM[N \% SERCOM\_Instance\_Number]$$

With N representing the LUT number and $i=0,1,2$ representing the LUT input index.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as input pad to the CCL.

Figure 37-9. SERCOM Input Selection



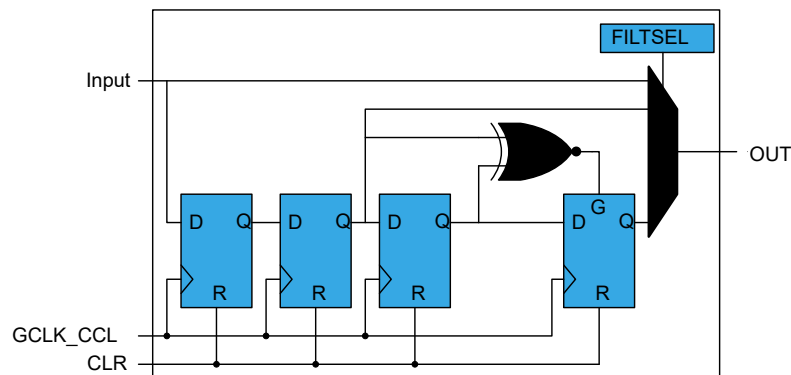
37.11.2.5 Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.

Note: Events used as LUT input will also be filtered, if the filter is enabled.

Figure 37-10. Filter



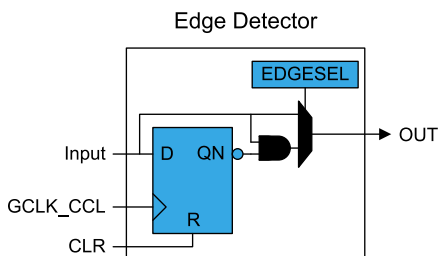
37.11.2.6 Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table should be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

Figure 37-11. Edge Detector



37.11.2.7 Sequential Logic

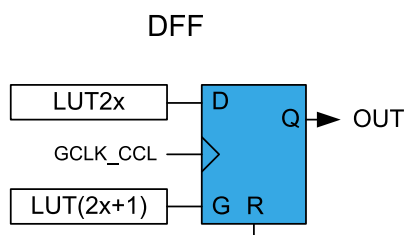
Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

Note: While configuring the sequential logic, the even LUT must be disabled. When configured the even LUT must be enabled.

Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 37-12. D Flip Flop



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

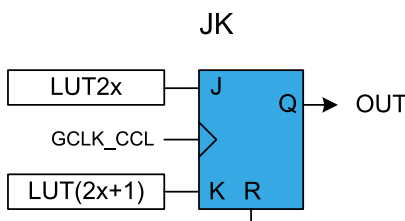
Table 37-2. DFF Characteristics

R	G	D	OUT
1	X	X	Clear
0	1	1	Set
		0	Clear
	0	X	Hold state (no change)

JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (LUT0 and LUT2), and the K-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 37-13. JK Flip Flop



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in the following table.

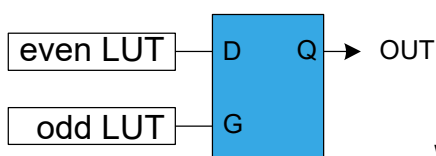
Table 37-3. JK Characteristics

R	J	K	OUT
1	X	X	Clear
0	0	0	Hold state (no change)
0	0	1	Clear
0	1	0	Set
0	1	1	Toggle

Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 37-14. D-Latch



When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

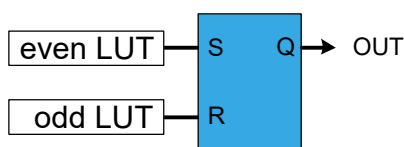
Table 37-4. D-Latch Characteristics

G	D	OUT
0	X	Hold state (no change)
1	0	Clear
1	1	Set

RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output (LUT0 and LUT2), and the R-input is driven by the odd LUT output (LUT1 and LUT3), as shown in the following figure.

Figure 37-15. RS-Latch



When the even LUT is disabled LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in the following table.

Table 37-5. RS-Latch Characteristics

S	R	OUT
0	0	Hold state (no change)
0	1	Clear
1	0	Set
1	1	Forbidden state

37.12 Events

The CCL can generate the following output events:

- LUTOUTn with n=0-3: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRLx.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The CCL can take the following actions on an input event:

- INSELY: The event is used as input for the TRUTH table.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRLx.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

37.13 Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in Standby Sleep mode.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

37.14 Peripheral Dependencies

Table 37-6. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
CCL0	0x4482E000		APB: CLKMSK[1] MASK20	PCHCTRL[36]	42 STATUS1 PERID10		<u>Generators:</u> LUTOUT0: CHANNEL[83] LUTOUT1: CHANNEL[84] LUTOUT2: CHANNEL[85] LUTOUT3: CHANNEL[86] <u>Users:</u> LUTIN0: USER[53],Resynch LUTIN1: USER[54],Resynch LUTIN2: USER[55],Resynch LUTIN3: USER[56],Resynch	PD_CORE_SW

Table 37-6. Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
CCL1	0x44830000		APB: CLKMSK[1] MASK21	PCHCTRL[37]	43 STATUS1 PERID11		<u>Generators:</u> LUTOUT0: CHANNEL[87] LUTOUT1: CHANNEL[88] LUTOUT2: CHANNEL[89] LUTOUT3: CHANNEL[90] <u>Users:</u> LUTINO: USER[57],Resynch LUTIN1: USER[58],Resynch LUTIN2: USER[59],Resynch LUTIN3: USER[60],Resynch	PD_CORE_SW

37.15 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRL	7:0		RUNSTDBY					ENABLE	SWRST
0x01	Reserved									
...										
0x03										
0x04	SEQCTRL0	7:0						SEQSEL[3:0]		
0x05	SEQCTRL1	7:0						SEQSEL[3:0]		
0x06	SEQCTRL2	7:0						SEQSEL[3:0]		
0x07	SEQCTRL3	7:0						SEQSEL[3:0]		
0x08	LUTCTRL0	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x0C	LUTCTRL1	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x10	LUTCTRL2	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x14	LUTCTRL3	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x18	LUTCTRL4	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x1C	LUTCTRL5	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x20	LUTCTRL6	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		
0x24	LUTCTRL7	31:24	TRUTH[7:0]							
		23:16		LUTEO	LUTEI	INVEI		INSEL2[3:0]		
		15:8	INSEL1[3:0]					INSEL0[3:0]		
		7:0	EDGESEL		FILTSEL[1:0]			ENABLE		

37.15.1 Control

Name: CTRL
Offset: 0x0
Reset: 0x00
Property: RW

Note: SEQCTRLn register is Enable-protected when CTRL.ENABLE = 1 or LUTCTRLn.ENABLE = 1.

Bit	7	6	5	4	3	2	1	0
		RUNSTDBY					ENABLE	SWRST
Access		RW					RW	W
Reset		0					0	0

Bit 6 – RUNSTDBY Run in Standby

Value	Name	Description
0	DISABLE	Generic clock is not required in standby sleep mode
1	ENABLE	Generic clock is required in standby sleep mode

Bit 1 – ENABLE Enable

Value	Name	Description
0	DISABLE	The peripheral is disabled
1	ENABLE	The peripheral is enabled

Bit 0 – SWRST Software Reset

Value	Name	Description
0	DISABLE	The peripheral is not reset
1	ENABLE	The peripheral is reset

37.15.2 SEQ Control x

Name: SEQCTRL
Offset: 0x04 + n*0x01 [n=0..3]
Reset: 0x00
Property: RW

Note: SEQCTRLn register is Enable-protected when CTRL.ENABLE = 1 or LUTCTRLn.ENABLE = 1.

Bit	7	6	5	4	3	2	1	0
					SEQSEL[3:0]			
Access								
Reset					0	0	0	0

Bits 3:0 – SEQSEL[3:0] Sequential Selection

These bits select the sequential configuration:
Sequential Selection

Value	Name	Description
0	DISABLE	Sequential logic is disabled
1	DFF	D flip flop
2	JK	JK flip flop
3	LATCH	D latch
4	RS	RS latch

37.15.3 LUT Control x

Name: LUTCTRL
Offset: 0x08 + n*0x04 [n=0..7]
Reset: 0x00000000
Property: RW

Note: LUTCTRLn register is Enable Protected when CCL.LUTCTRLn.ENABLE = 1.

Table 37-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRUTH[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
		LUTEO	LUTEI	INVEI	INSEL2[3:0]			
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	INSEL1[3:0]				INSEL0[3:0]			
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	EDGESEL		FILTSEL[1:0]				ENABLE	
Access	RW		RW	RW			RW	
Reset	0		0	0			0	

Bits 31:24 – TRUTH[7:0] Truth Value

These bits define the value of truth logic as a function of inputs IN[2:0].

Bit 22 – LUTEO LUT Event Output Enable

Value	Name	Description
0	DISABLE	LUT event output is disabled
1	ENABLE	LUT event output is enabled

Bit 21 – LUTEI LUT Event Input Enable

Value	Name	Description
0	DISABLE	LUT incoming event is disabled
1	ENABLE	LUT incoming event is enabled

Bit 20 – INVEI Inverted Event Input Enable

Value	Name	Description
0	NORMAL	Incoming event is not inverted
1	INVERTED	Incoming event is inverted

Bits 19:16 – INSEL2[3:0] Input Selection 2

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	PERSEL0	Peripheral 0 input source
0x6	PERSEL1	Peripheral 1 input source
0x7	PERSEL2	Peripheral 2 input source
0x8	PERSEL3	Peripheral 3 input source
0x9	PERSEL4	Peripheral 4 input source
0xA	PERSEL5	Peripheral 5 input source
0xB	ASYNCEVENT	Asynchronous event input source. The EVENT input will bypass edge detection logic.

Bits 15:12 – INSEL1[3:0] Input Selection 1

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	PERSEL0	Peripheral 0 input source
0x6	PERSEL1	Peripheral 1 input source
0x7	PERSEL2	Peripheral 2 input source
0x8	PERSEL3	Peripheral 3 input source
0x9	PERSEL4	Peripheral 4 input source
0xA	PERSEL5	Peripheral 5 input source
0xB	ASYNCEVENT	Asynchronous event input source. The EVENT input will bypass edge detection logic.

Bits 11:8 – INSEL0[3:0] Input Selection 0

Value	Name	Description
0x0	MASK	Masked input
0x1	FEEDBACK	Feedback input source
0x2	LINK	Linked LUT input source
0x3	EVENT	Event input source
0x4	IO	I/O pin input source
0x5	PERSEL0	Peripheral 0 input source
0x6	PERSEL1	Peripheral 1 input source
0x7	PERSEL2	Peripheral 2 input source
0x8	PERSEL3	Peripheral 3 input source
0x9	PERSEL4	Peripheral 4 input source
0xA	PERSEL5	Peripheral 5 input source
0xB	ASYNCEVENT	Asynchronous event input source. The EVENT input will bypass edge detection logic.

Bit 7 – EDGESEL Edge Selection

Value	Name	Description
0	DISABLE	Edge detector is disabled
1	ENABLE	Edge detector is enabled

Bits 5:4 – FILTSEL[1:0] Filter Selection

These bits select the LUT output filter options:
Filter Selection

Value	Name	Description
0	DISABLE	Filter disabled
1	SYNCH	Synchronizer enabled
2	FILTER	Filter enabled

Bit 1 - ENABLE LUT Enable

Value	Name	Description
0	DISABLE	LUT block is disabled
1	ENABLE	LUT block is enabled

38. Analog to Digital Converter (ADC)

38.1 Overview

The Analog-to-Digital Converter can convert one independent signal using the SAR (Successive Approximation Register) ADC Module. The ADC module supports up to 12 bit resolution. For sample timing and conversion rates, see the Analog-to-Digital Converter (ADC) Electrical Specifications. Single-ended and differential measurements are supported. Signed and unsigned sampling formats are supported.

The SAR ADC module can support up to 16 separate analog input channels.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal AVDD and external VREFH reference voltages can be used (i.e., AVDD or VREFH respectively).

The ADC has a digital result compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required. Hardware oversampling is also supported in the ADC's hardware.

Note: If $AVDD < 2.5V$, the user must enable analog charge pumps in SUPC.VREGCTRL.CPEN[1:0].

38.2 Features

Up to 12-bit resolution of the numerical output, signed or unsigned (Higher resolutions possible with oversampling).

- Signed/Unsigned results
- Left or Right aligned result
- Fractional or integer results
- External analog inputs:
 - Up to 12 single ended input channels (AIN[11:0])
 - Or up to 3 differential inputs channels (AIN0/AIN1, AIN2/AIN3, AIN4/AIN5), plus 6 single ended input channels (AIN[11:6])
- Internal inputs:
 - VDDCORE internal analog channel AIN12
 - IVREF 1.2v, internal analog channel AIN14
- Up to 16 trigger sources, off-chip hardware or on-chip hardware or software generated per analog input channel.
- Edge or level active triggering modes, generating single conversions or bursts of conversions.
- A scan trigger to start a scan cycle which can individually include or not include any of the analog inputs.
- Any of the 16 trigger sources or the scan triggers can be assigned to individual analog input channels.
- The scan trigger itself can select any of the 16 trigger sources as its own source.
- Programmable sampling time, CORCTRL.SAMC.
- Each analog input/channel output register can be read from a general dedicated output register (Write to the CORDYID and CHRDYID registers and then reads the CHRDYDATA register).
- 16sample deep FIFO supporting all channels.

- Digital Comparator for monitoring output values in relation to user-specified ADC result threshold.
- Digital filter; providing averaging/oversampling for increased noise immunity and are assignable to any analog input.

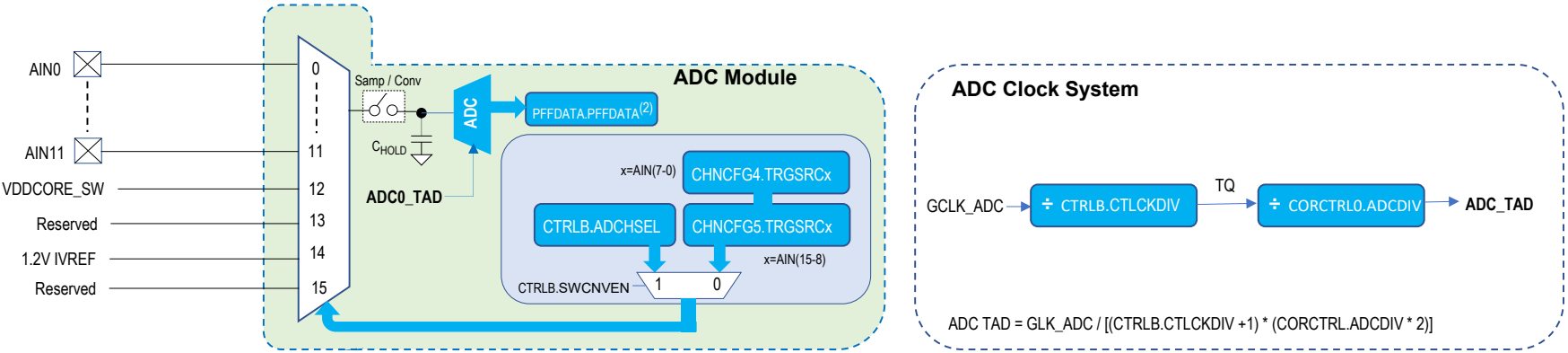
38.3 Block Diagram

The ADC Module has a dedicated Analog Input Multiplexer, which can select from seven up to 16 different analog inputs. The ADC Module has a Post Processing Block, consisting of a dedicated Digital Filter and a dedicated Digital Comparator.

The ADC's sample data is available in the ADC Channel Ready Data Register (CHRDYDAT) and a 16-sample-deep FIFO (PFFDATA).

The ADC module provides two interrupts to the NVIC: one Global Interrupt and one ADC Module-specific interrupts. Global interrupts are serviced by the CTLINTENCLR, CTLINTENSET, and CTLINTFLAG registers, while the ADC module's interrupts are serviced by INTENCLR, INTENSET, and INTFLAG registers.

Figure 38-1. ADC Block Diagram



The ADC runs on a peripheral clock provided by the Generic Clock (GCLK) module, known as the GCLK_ADC, and identified as ADC Control Clock, or CTL_CLK in this chapter. The ADC Module has a clock derived from the ADC Control Clock, CORE_CLK. ADC Special Function Registers (SFRs) are identified as to which clock domain they belong: "APB_CLK" for the APB/Main Clock and "GCLK" for the clock derived from the GCLK, i.e. the ADC Control Clock (CTL_CLK).

The ADC Data Bus is 19 bits wide: { ChannelValid, CoreChannelID[5:0], ChannelData[11:0] }. The ChannelValid signal identifies whether the core's time slot contains valid data. The input channel index is reported in the CoreChannelID bits. The captured signal is reported in the 12 bits of ChannelData.

The ADC Data Bus sends data to the ADC's status and data registers as well as to the Digital Filter and Digital Comparator Post Processing Blocks. The bus also supports sending filtered results to the ADC's data registers.

38.4 Signal Description

Table 38-1. Signal Description

Signal	Type	Description
AVDD	Power	Analog Power Supply
AVSS	Power	Analog Ground Supply
ADC_VREFH	Reference	External voltage reference
ADC AINx, x = 0,1,...,11	Analog	AINx Analog inputs for ADC Module 0

Notes:

1. Channel 12 is attached internally to the Temperature Sensor Monitor.
2. Channel 13 is attached internally to the IVREF, Band Gap 1.2 Volt Monitor.

Note: Please consult the section "Signal Description" and "ADC Pinout I/O Description" table to determine which pins provide which input signals, depending on the device package in use.

All the AINx signals listed above can be used for single-ended input and/or 3 for differential inputs pairs.

- (ADC_AIN0 (Diff +) , ADC_AIN1 (Diff -))
- (ADC_AIN2 (Diff +) , ADC_AIN3 (Diff -))
- (ADC_AIN4 (Diff +) , ADC_AIN5 (Diff -))

These pairs are configured in the CHNCFG3n (n=1,2,3) register by setting the DIFF bits corresponding to the signal pairs in use.

38.5 Functional Description

38.5.1 Indexing

There are four ADC Modules, each of which has varying numbers of input channels. Each ADC has a dedicated Digital Filter and a dedicated Digital Comparator:

- There are four ADCn modules : $n = 0,1,2,3$
- There are four dedicated Digital Filters are indexed by $n : n = 0,1,2,3$
- There are four dedicated Digital Comparators are indexed by $n : n = 0,1,2,3$
- Each of the four ADC modules that have a varying number of input signals:
ADC0: External AIN[15:0] inputs
ADC1: External AIN[5:0] inputs and internal AIN6=VDDCORE
ADC2: External AIN[5:0] inputs and internal AIN6=Temperature Sensor
ADC3: External AIN[5:0] inputs and internal AIN6=1.2v IVREF

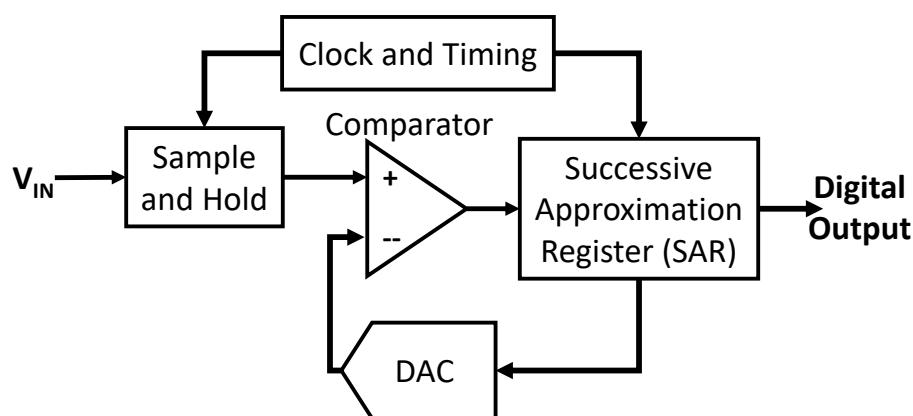
38.5.2 Principle of Operation

➔ Important: When $AVDD < 2.5V$, for proper ADC operation it is critical that the internal charge pumps be enabled as necessary as define in SUPC.VREGCTRL.CPEN[1:0].

The basic architecture of an ADC Module is shown in the following figure.

At the start of a capture cycle the Sample and Hold capacitor is connected to the incoming voltage on a trigger event until the expiration of the sample time defined by CORCTRLx.SAMC. At the end of the sampling period the Sample and Hold is disconnected from the input signal and connected to the Comparator and the conversion sequence begins. ADC conversion time = $(\# \text{ bits} + 1) \cdot TAD$.

Figure 38-3. ADC Module Architecture



38.5.3 Basic Operation

38.5.3.1 Enabling, Disabling, and Resetting

The ADC is enabled by setting **CTRLA.ENABLE** = 1. Setting this bit to zero disables the ADC. The ADC can be reset by setting **CTRLA.SWRST** = 1 to initiate a software reset. (The ADC module is reset when the **SYNCRBUSY.SWRST** bit goes low.)

Note: The bit **DBGCTRL.DBGRUN** is unchanged by a software reset.

38.5.3.2 Operation

A trigger event initiates the start of the ADC sampling of the selected analog input. After the expiration of the SAMC sample time the conversion sequence is automatically initiated by the hardware assuming CTRLB.SWCNVEN = 0. For maximum efficiency, in the case of SCAN mode, at the conclusion of sampling for the current active analog input and while the conversion cycle is in progress, the analog channel input multiplexer will switch to the next analog input in the users scan sequence in preparation to initiate the next channel sampling sequence immediately once the previous channel conversion is complete. In SCAN mode, sample/conversion progression is always lower to higher numbered channels as selected by the user in the CHNCFG2.CSSx register.

38.5.3.3 Clocks Setup

The interface to the Advanced Peripheral Bus (APB) is clocked by the system's Main Clock. The main clock (GCLK0 or MCLK) can operate up to 120 MHz.

The ADC Core operates on a clock provided by the GCLK module, GCLK_ADC max 100Mhz. The register field GCLK.PCHCTRL33.GEN[2:0] is configured to select one of the GCLK's clock generators. Clock generators are configured by the register GCLK.GENCTRLm, for m=0,1,...,11. This register

selects the input clock for the generator and provides the clock divider to divide an input clock down to the clock output of GCLK Generator m . This clock is called the ADC Control Clock, or CTL_CLK in this chapter.

38.5.3.4 Conversion Timing and Sampling Rate

If the period of CTL_CLK is T_Q , then the period of the respective ADC_CLK module clock, T_{AD} , is given by:

$$T_{AD} = 2 \cdot \text{ADCDIV} \cdot T_Q .$$

The minimum ADC clock period is $T_{AD} = 2 \cdot T_Q$.

The maximum ADC throughput rate with N_{BITS} of resolution, is:

Provided CTRLB.SWCNVEN=0:

Maximum Throughput rate $F_{TPR} = [1 / [(\text{CORCTRL.SAMC}+2)T_{AD} + (N_{BITS}+1)T_{AD}]] / \# \text{Active ADC Channels}$

Example:

(CORCTRL.SAMC=0x1, 12bit resolution and 2 AINx scan channels selected)

$$F_{TPR} = [1 / (3T_{AD} + 13T_{AD})] / 2 \\ = (1 / 16T_{AD}) / 2$$



Important: The TAD clock must have the following characteristics:

1. $TAD \leq 75 \text{ MHz}$
2. The highest value possible given the integer divisor relationship with the clock source

Examples are as follows:

- Clock Source TAD CLOCK(min.):
 - 300 MHz to 75 MHz
 - 200 MHz to 50 MHz
 - 150 MHz to 75 MHz
 - 120 MHz to 60 MHz
 - 72 MHz to 72 MHz

38.5.3.5 Voltage Reference Selection

The ADC's voltage reference is selected by CTRLD.VREFSEL[2:0]:

VREFSEL[2:0]	ADREF+	ADREF-
000	A_{VDD}	A_{VSS}
001	External V_{REF+H}	A_{VSS}
010 - 111	Reserved	Reserved

38.5.3.6 ADC Resolution

The output resolution is selected by CORCTRL.SELRES, to choose between 6 bits, 8 bits, 10 bits, or 12 bits (default). Changing the resolution of the ADC module will also change the ADC throughput rate.

Note: Required analog signal sample time in ADC clock cycles (TAD) as defined in CORCTRL.SAMC depend on source impedance of external signal source. Consult electrical specifications for required sample time based on external source impedance.

38.5.3.7 Differential and Single Ended Conversions

The ADC can measure up to 12 single ended input channels (AIN[11:0]), or up to 3 differential inputs channels (AIN0/AIN1, AIN2/AIN3, AIN4/AIN5), plus 6 single ended input channels (AIN[11:6]).

Refer to the CHNCFG3 register for bit descriptions on how to configure them for differential mode.

38.5.3.8 Input Channel Conversion Priority

The selection of the next analog input for conversion uses a natural ascending analog input channel order priority. In scan mode, analog input channels are always processed lower to higher channels as selected by the user in the ADC CHNCFG2n.CSS register.

38.5.3.9 Output Data Formats

The Analog/Digital core provides up to 12-bits of resolution which are available in registers CHRDYDAT and PFFDATA as 16 bits. The 12-bit ADC output data is translated into a 16-bit format specified by CHNCFG2.FRACK. Additionally, the CHNCFG2.SIGNk bit specifies the data format as unsigned, or two's complementary signed.

The following table shows the four cases of the channel index k 16-bit output data format depending on the values [FRACK ,SIGNk]:

Table 38-2. ADC Analog Input Channel n Data Format Examples

FRACK	SIGNk	16-bit Output Data Format
-	-	s = sign bit d = mantissa value bit
0	0	0000 dddd dddd dddd
0	1	ssss sddd dddd dddd
1	0	dddd dddd dddd 0000
1	1	sddd dddd dddd 0000

38.5.3.10 Initialization

The following steps are required to configure the ADC for operation:

1. Configure input pins for operation as analog input pins, ADC_AINK.
2. Configure a GCLK generator to provide the GCLK_ADC clock (PCHCTRLm index = 33).
3. Configure ADC registers:
 - a. Copy factory-provided calibration values from the CAL OTP Register FCCFG65 into CALCTRL for all the ADC in use by the application.
 - b. Configure CTRLA as needed but do not set CTRLA.ENABLE.
 - c. Configure CTRLD, choosing the clock divider (CTLCKDIV) for the CTL_CLK clock based on the GCLK_ADC, Voltage Reference (VREFSEL), and Wake-up Delay Exponent (WKUPEXP).
 - d. (Optionally) Enable ADC interrupts for FIFO support using CTLINTENSET. Optionally enable the interrupts for VREFRDY and Core n ready. (Or these conditions can be monitored by watching CTLINTFLAG in a while(1) loop.) Enable interrupt vector 132 (ADC Global Interrupt) in the NVIC.
 - e. Enable ADC interrupts using the INTENSET registers. Enable the corresponding NVIC interrupt vector 133.
 - f. For each ADCn, configure its Control register CORCTRLn:
 - Set the clock divider, ADCDIV, for the ADC Module clock, CORE_CLK[n].
 - Set the sample count SAMC (ADCDIV and SAMC determine the ADC Module's sample rate fs).
 - If ADC resolution other than 12 bits is desired, set it using SELRES.
 - Other settings in CORCTRLn as needed.

- g. Setup differential channels using $\text{CHNCFG3n.DIFFk} = 1$ for signals pairs k and $k+1$ (k even).
- h. Select the input channels for each scan using $\text{CHNCFG2n.CSSk} = 1$.
- i. Set the data format for each sample using CHNCFG2n.FRACTk and CHNCFG3n.SIGNk .
- j. Setup channel triggers.
 - (Optional: Setup up the Digital Filter and Digital Comparator associated with each of the four modules).
4. Enable the ADC by setting $\text{CTRLA.ENABLE} = 1$.
5. Wait until $\text{CTLINTFLAG.VREFRDY} = 1$, either using an ISR or polling in a $\text{while}(1)$ loop. This signals that the chip's analog environment is ready.
6. Wait for CRRDYn to go high in CTLINTFLAG , indicating that the corresponding ADC Module is ready. This can be polled in a $\text{while}(1)$ loop or trapped by the ISR attached to the ADC Global Interrupt.

38.5.3.11 Channel Triggers

The ADC controller starts the capture and conversion of an input channel when that channel is “triggered”. The ADC Controller needs to know when a channel requests an analog-to-digital conversion. For this purpose, each input channel contains a trigger signal which is passed by the channel to the Module's Access Arbiter. The Access Arbiter drives the address selections to both the input Analog Multiplexer and to the output Digital De-multiplexer.

38.5.3.11.1 Trigger Priority

When two triggers arrive at the Access Arbiter at the same time the trigger associated with the input channel having the lower index k will be serviced. The second trigger will be queued for later service, provided it is not overwritten by a third trigger request. In that case, the third trigger will be lost and not serviced.

38.5.3.11.2 Channel Trigger Selection

The trigger used for the ADC, Input Channel k , is specified by $\text{CHNCFG}\{4|5\}\text{n.TRGSRCK}[3:0]$. (CHNCFG4n contains the trigger choices for $k = 0, 1, \dots, 7$. CHNCFG5n contains the trigger choices for $k = 8, 9, \dots, 15$. TRGSRCK for $\text{Sn} \leq k$ are not defined.)

The possible values for TRGSRCK are:

- = 0000: No Trigger (NOP)
- = 0001: Global Software Trigger (CTRLB.GSWTRG)
- = 0010: Global Level Software Trigger (CTRLB.LSWTRG)
- = 0011: SCANTRG - Scan Trigger
- = 0100: STRIG Synchronous Trigger
- = 0101 - 1111: ADC Trigger Event User 0 – 10

Input channels with no triggers will not be serviced during the Module's operation. For all trigger sources except the SCANTRG, setting the TRGSRCK will result in only input channel k being serviced by the Module. To collect a “scan” of input channels TRGSRCK must be set to SCANTRG and then SCANTRG is defined by CORCTRLn.STRGSRC . All the input channels to be included in a scan started by SCANTRG must have CHNCFG2n.CSSk set to one.

The SCANTRG source is selected by CORCTRLn.STRGSRC :

- = 0000: No trigger (NOP)
- = 0001: Global Software trigger (CTRLB.GSWTRG)
- = 0010: Level Software trigger (CTRLB.LSWTRG)
- = 0011: Reserved

= 0100: Synchronous Trigger (STRIG)

= 0101 - 1111: EVSYS Event Generator 0 – 10

The Synchronous Trigger (STRIG) is driven by a counter at the ADC Control Clock (CTL_CLK) frequency and fires when the counter reaches the value defined by CTRLC.CNT[7:0]. To enable this clock set CTRLB.STRGEN to one.

38.5.3.11.3 Software Triggers

ADC captures can be directly controlled from software by using the Global Software Trigger to start a single capture (when CTRLB.GSWTRG = 1) or by using the Global Level Software Trigger to start a burst of captures that will continue as long CTRLB.LSWTRG = 1 and stop when LSWTRG = 0.

38.5.3.11.4 ADC Debugging

Setting CTRLB.SWCNVEN = 1 allows two bits in CTRLB to control the Module and the input channel specified by CTRLB.ADCHSEL. For this to work all the other input channels for the Module must be disabled by setting TRGSRCk = 0, which can only be accomplished when the ADC is disabled. Sampling of the specified input channel starts when CTRLB.SAMP is set to one and stops when CTRLB.SAMP is reset to zero. Immediately after this start conversion by setting CTRLB.RQCNVRT to one.

The CTRLB register has a dedicated SYNCBUSY bit to allow the manipulation of these bits when the ADC is enabled. After writing to the CTRLB register the SYNCBUSY.CTRLB will go high. Wait until the bit goes low before writing to CTRLB again.

38.5.3.11.5 Trigger Limitations

Trigger Rule 1: In order to ensure synchronizing every single pulse on GSWTRG, the GSWTRG pulses must be at least 4 ADC Control Clock periods apart, positive edge to positive edge (because the GSWTRG lasts only 1 single APB clock period by construction).

Trigger Rule 2: If a channel k is effectively included in scan n by setting CHNCFG{4|5}.n.TRGSRCk = 3 (SCANTRG) and CHNCFG2n.CSSk = 1, then the user MUST ensure that NO other triggers are generated for that channel using CTRLB.RQCNVRT or any digital filter. Otherwise the scan behavior is unpredictable.

Trigger Rule 3: In order to ensure synchronizing every single pulse on LSWTRG, the LSWTRG pulses must be at least 8 ADC Control Clock periods + 10 Main Clock clock periods positive edge to positive edge, and also at least 4 ADC control clock periods + 4 Main Clock periods negative edge to positive edge.

38.5.4 ADC Result Registers

Each of the 14 analog input channels has an associated results register. To read the converted data write the ADC Module index n and the channel index k in the CORCHDATAID register:

CORCHDATAID.CORDYID = n

CORCHDATAID.CHRDYID = k

Then read the data from the CHRDYDAT register. The captured data bits are stored in CHRDYDAT, together with the FRACT, SIGN, DIFF, and LVL settings associated with this input channel. CHRDYDAT is 16 bits wide. For more information on the data formats found in this field consult *Output Data Formats*.

When data from ADCn, input channel k, is ready to be read by the application the INTFLAGn.CHRDY[k] bit will be set. If the corresponding bit in INTENSETn has been set, then the interrupt associated with ADCn will fire. (To reset the CHRDY bit write one to it.) If the application does not read the new data before another sample arrives the Channel Overwritten Error flag (INTFLAGn.CHNERRC) bit will be set.

38.5.5 ADC Result FIFO

The ADC has a 16-sample deep FIFO. To capture ADCn results in this FIFO, first enable the FIFO: PFFCTRL.PFFEN = 1. Then enable the FIFO PFFCTRL.PFFCR0 = 1. FIFO results are available in the PFFDATA register.

The status of the FIFO is reflected in the CTLINTFLAG register. Setting the corresponding bits in the CTLINTENSET register enables these bits to fire the ADC Global Interrupt.

38.5.6 Additional Features

38.5.6.1 Interleaving Samples for Higher Sample Rate

The interleaving of ADC COREs provides a method of increasing the sampling rates of the ADC. The interleaving process involves two or more ADC cores (determined by CTRLC.CORINTERLEAVED[1:0]) with a given same analog input signal being sampled at different times by the interleaved ADCs. Only one input channel per each ADC module must be configured for the interleaving trigger source. The remaining input channels must be configured for NOP triggers. The trigger mode must be EVSYS rising edge only, not level.

Table 38-3. Interleaved ADCs

Number of Interleaved ADCs Used	Interleaved ADC Modules	CTRLC.CORINTERLEAVED	12-bit Msps (Max.) [Min Trigger rate]	10-bit Msps (Max.) [Min Trigger rate]	8-bit Msps (Max.) [Min Trigger rate]	6-bit Msps (Max.) [Min Trigger rate]
2	1,2	=0b001	1 / (8 * TAD) [8 TAD]	1 / (7 * TAD) [7 TAD]	1 / (6 * TAD) [6 TAD]	1 / (4.5 * TAD) [4.5 TAD]
	0,1	=0b100	1 / (10 * TAD) [10 TAD]	1 / (8.5 * TAD) [8.5 TAD]	1 / (7.5 * TAD) [7.5 TAD]	1 / (6.5 * TAD) [6.5 TAD]
3	1,2,3	=0b010	1 / (5.5 * TAD) [5.5 TAD]	1 / (5 * TAD) [5 TAD]	1 / (4 * TAD) [4 TAD]	1 / (3 * TAD) [3 TAD]
	0,1,2	=0b101	1 / (6.5 * TAD) [6.5 TAD]	1 / (6 * TAD) [6 TAD]	1 / (5 * TAD) [5 TAD]	1 / (4.5 * TAD) [4.5 TAD]
4	0,1,2,3	=0b011	1 / (5 * TAD) [5 TAD]	1 / (4.5 * TAD) [4.5 TAD]	1 / (4 * TAD) [4 TAD]	1 / (3.5 * TAD) [3.5 TAD]

Minimum CORCTRLn.SAMC sample time values for 12/10/8/6-bit Resolution:

- **ADC0:** CORCTRL0.SAMC = 4 (i.e., = 6 TAD sample time)
- **ADC1:** CORCTRL1.SAMC = 1 (i.e., = 3 TAD sample time)
- **ADC2:** CORCTRL2.SAMC = 1 (i.e., = 3 TAD sample time)
- **ADC3:** CORCTRL3.SAMC = 1 (i.e., = 3 TAD sample time)
- Conversion Time = (#bits Resolution + 1)



Important: When interleaving ADC modules, users must use the same worst case CONFIG[n].CORCTRL.SAMC value of the slowest ADC for all the active interleaved ADC's SAMC values. All ADC must use the same SAMC sample time. In this case ADC0, 6 TAD is the minimum sample time and ADC1/2/3 must be the same if used together in interleaved mode for the fastest configuration.

Notes:

1. The table above assumes event trigger source GCLK is 2x GCLK_ADC. This allows 0.5 TAD increments. If not, and trigger source GCLK is equal to GCLK_ADC then maximum throughput rate and minimum trigger rate must be rounded up to next whole integer TAD value.
2. TAD is the ADC_CLOCK period time in nano seconds (see ADC electrical specifications).

$$\text{ADCn TAD} = \text{GLK\_ADC} / [(\text{CTRLD.CTLCKDIV} + 1) * (\text{CONFIG}[n].\text{CORCTRL.ADCDIV} * 2)]$$
3. Assumes EVSYS trigger peripheral clock = $(2 / \text{TAD}) = (2 * \text{GCLK\_ADC})$.
4. Must use the same analog input AINx on ADCn modules being interleaved.
5. These bits are Enabled Protected (writes are ignored when CTRLA.ENABLE = 1 and will return a bus error).
6. ADC0, due to the higher number of analog inputs it services have a higher minimum CONFIG[n].CORCTRL.SAMC sample time that the user must consider since the slowest ADC affects the maximum combination of ADC interleaved throughput rate. ADC 1, 2, and 3 have an identical CONFIG[n].CORCTRL.SAMC minimum sample time.
7. In interleaved mode, for the user selected active interleaved ADC modules, the sample or conversion sequence occurs from lowest-to-highest ADC modules (natural order priority ADC0 -> ADC1 -> ADC2 -> ADC3). The trigger event, although it must be common to all the interleaved ADC modules, is sequenced by the ADC hardware singularly, one at a time, to each interleaved ADC group according to the natural priority of the ADC modules.
8. In interleaved mode for fastest conversion speeds, the trigger clock source timing resolution should be TAD/2.
 For example, ADC triggered by TCC timer.
 $\text{ADC TAD} = 75 \text{ MHz}, (13.333 \text{ ns}), \text{TCC timer increment} = 150 \text{ MHz}, (6.667 \text{ ns})$



Important: All ADCs used in an ADC interleaved group must use the same CONFIG[n].CORCTRL.ADCDIV and CORCTRLn.SAMC settings, and the same singular peripheral for the Event System (EVSYS) and ADC trigger source to minimize clock skew and phase shifts between ADC modules to maintain a consistent and coherent sample or conversion timing between all the linked interleaved ADC's.

38.5.6.2 Dedicated Digital Filter

The Dedicated Digital Filter enables the ADC Module to feed a contiguous (i.e. back-to-back) set of input samples into a digital filter to produce a single finished output sample that either “averages” or “oversamples” the input samples. All the captured samples are equally weighted in the accumulation of the result before scaling of the final result.

The main difference between the “averaging” and “oversampling” modes relates to how the final filter output is scaled. Collecting 16 samples in “oversampling” mode generates two additional bits of output (14 bits total) while in “averaging” mode the output is always exactly 12 bits wide for all burst sizes.

The Digital Filter trades ADC throughput for increased data output resolution, in the sense that if random noise contaminates the least significant bits of a sample then “oversampling” can increase the resolution of the results. Oversampling by a factor of N increases the number of effective bits (ENOB) by square root(N). The scaling in “oversampling” mode for small N conserves all the data but for larger N some resolution is lost in order to fit the filters results into just 16 bits.

On the other hand, if the noise contamination of the samples is not random, but is harmonically related to another signal, this filtering will not improve ADC performance (ENOB does not change.).

38.5.6.2.1 Operation of the Digital Filter

The Digital Filter must be enabled by setting FLTCTRL.FLTEN. The input channel to be filtered, with index k , is set with FLTCTRL.FLTCHNID = k . The choice between “averaging” or “oversampling” is implemented by FLTCTRL.FMODE.

Once the Digital Filter is configured, the filter’s control logic waits for an external trigger of channel k to initiate the contiguous scan of samples defined by FLTCTRL.OVRSAM. The trigger signal for the channel k to be filtered causes the accumulator to be cleared and initiates the first conversion.

On being enabled, the filter control logic has forced the trigger sensitivity into Level Mode (by logic appended to the internal registers associated with the CHNCFG1.LVN k registers), which enables the ADC to work in burst mode collecting back-to-back samples. The filter control logic will force internally the trigger to one as long as the filter needs more samples. In this way the ADC itself will collect a burst of samples spaced apart according to CORCTRL.SAMC.

When the required number of samples have been received and processed by the filter logic, the filter control logic releases the forcing of the channel trigger and starts waiting for a new initial sample (when the external trigger determines the sampling event according to the trigger waveform and the trigger setting to edge / level sensitivity).

At the completion of a burst collection the filter control logic will set the INTFLAG.FLTRDY flag, which will trigger the ADC interrupt if the corresponding bit in INTENSET has been set. Then the new output will be available using the CORCHDATAID and CHRDYDAT registers as described in ADC Result Registers.

38.5.6.2.2 Digital Filters Usage

To sustain the back-to-back collection of input samples the digital filter module will block any lower priority ADC conversion requests until the required input samples have been completed. However, any higher priority ADC requests will be processed un-impeded. Therefore, if higher priority requests occur during the filter sequence, the collected samples will no longer be contiguous, injecting unacceptable levels of aperture jitter over the burst and therefore contaminating the filter output. *For this reason, the user should arrange the initiation trigger for the over burst collection to occur while there are no expected interruptions from higher priority ADC conversion requests.*

The register field FLTCTRL.FMODE determines the filtering mode. For “averaging” mode, FMODE=1. For “oversampling” mode, FMODE=0. When FMODE=1, the register field FLTCTRL.DATA16EN controls the format of the filter’s output:

- FMODE=1 and DATA16EN=1, then the filter index output data is always left justified with all 16 bits significant.
- If FMODE=1 and DATA16EN=0, then the filter output data is always left justified but with only the first 12 bits significant and the last 4 bits always zero.

If FMODE=0 (i.e. when the Digital Filter works in accumulation mode), DATA16EN bit has no effect, but CHNCFG2.FRACT k and CHNCFG3.SIGN k determines the format of the filter output data as shown in Output Data Formats.

38.5.6.2.3 Digital Filters and the APB Bus FIFO

If PFFCTRL.PFFCR is set for the ADC using its Digital Filter the FIFO will capture all the back-to-back samples fed into the Digital Filter, save for the last scanned sample, which will be replaced by the filter’s output.

38.5.6.2.4 Finite Bursts

If the trigger of a burst collection is not repeated, then the ADC will collect just one burst. As long as the number of samples defined in OVRSAM is less than 16 then the samples collected will not overflow the FIFO. The first 15 will be directly available from the FIFO and, while the 16 sample is overwritten by the filter, the 16 sample can be recovered using the first 15 samples and the filter’s output.

38.5.6.3 Dedicated Digital Comparator

The ADC module has a dedicated Digital Comparator that can alert the application to the capture and conversion of input signals that match predefined criterion(s). When an input conversion produces a “hit” the flag INTFLAG.CMPHIT is set, which can optionally fire the interrupt if INTENSET.CMPHIT is set. The index, *k*, of the input channel that produce the “hit” is identified in the INTENSET.CMPINTID[3:0] field.

The Digital Comparator must be enabled by setting CMPCTRL.CMPEN. Only those input channels that have the associated CHNCMPEN_k bit set are included in the comparator scans.

The CMPCTRL register supports these comparisons directly:

- **bit 29 IEHIHI: Enable Comparison - High Limit, Active High:**
Setting this bit enables comparison events $ADCMPHI \leq \text{ADC value}$
- **bit 28 IEHILO: Enable Comparison - High Limit, Active Low:**
Setting this bit enables comparison events $\text{ADC value} < ADCMPHI$
- **bit 15 IEBTWN: Enable Comparison - Active Between Limits:**
Setting this bit enables comparison events $ADCMPO \leq \text{ADC Value} < ADCMPHI$
- **bit 14 IELOHI: Enable Comparison - Low Limit, Active High:**
Setting this bit enables comparison events $ADCMPO \leq \text{ADC Value}$
- **bit 13 IELOLO: Enable Comparison - Low Limit, Active Low:**
Setting this bit enables comparison events $\text{ADC Value} < ADCMPLO$

Since setting more than one of the IE* bits produces a condition that is the and of all conditions enabled, it is not a recommended configuration.

38.5.7 DMA Operation

Table 38-4. DMA Event Trigger Mapping

Instance	Channel	CHCTRLB.TR IG [index]	Presentation
ADC	PFFRDY	37	ADC DMA PFFRDY trigger

The DMA can be programmed to read the APB Bus FIFO when there is available data. The DMA can read from the FIFO when the FIFO is half full or when the FIFO is not empty (i.e. it is “Ready”). The bit **PFFCTRL.PFFRDYDMA** controls this choice.

For the FIFO to work, first enable bit by setting **PFFCTRL.PFFEN** to one. Next enable the FIFO to capture data from the ADC Module in operation by setting **PFFCTRL**.

38.5.8 Interrupts

The ADC module provides two interrupts to the NVIC: one Global Interrupt (REQ0) and one Module-specific interrupts (REQ1). Global interrupts are serviced by the **CTLINTENCLR**, **CTLINTENSET**, and **CTLINTFLAG** registers, while each ADC Module’s interrupts are serviced by **INTENCLR**, **INTENSET**, and **INTFLAG** registers.

When the ADC’s chosen voltage reference status changes (not ready to ready or the reverse) the **CTLINTFLAG.VREFUPD** bit will be set. For this bit to fire the global interrupt **CTLINTENSET.VREFUPD** should be set to one. The ADC’s global interrupt ISR should always check this flag and disable ADC operation when **CTLINTFLAG.VREFUPD** = 1 and **CTLINTFLAG.VREFRDY** = 0, indicating that the voltage reference is no longer “ready” and therefore no longer useable.

When the startup delay for each ADC Module has elapsed a **CRRDY_n** bit in the **CTLINTFLAG** register will go high. This can trigger a global interrupt as well if the corresponding bit in **CTLINTENSET** has been set. Note that the **CRRDY_n** bit in INTFLAG will stay high and therefore continue to fire the

global interrupt unless the global interrupt service routine disables the Core Ready Interrupts from firing by setting the **CTLINTENCLR.CRRDY n** bits to one before exiting.

Other bits in the **CTLINTFLAG** register are related to the operation of the APB Data FIFO. The **CTLINTFLAG.PFFHFUL** bit goes high when the FIFO is half full. The **CTLINTFLAG.PFFRDY** goes high when the FIFO is “ready”, i.e. when it contains data to be ready. FIFO overflow and underflow will set the **CTLINTFLAG.PFFOVF** or **CTLINTFLAG.PFFUNF** bits, respectively.

38.5.9 Events

The ADC can generate events, which can be shared by the Event System (EVSYS) module with other modules, and it can be an event user of events generated by other modules and distributed by the EVSYS module. The ADC can generate an event when a Digital Comparator “Hit” occurs. It can also generate an event when a new data result is “ready” (available for reading.) As a user the ADC often uses events generated by a timer to trigger the capture and conversion of new samples.

There is one Results Ready (ADC_RESRDY) generators. Similarly, there is one Comparator Hit (ADC_CMP) generators. The ADC has eleven “trigger” users in the EVSYS (TRIGN, $n = 0,1,...,10$). This provides eleven separate event users which can be used to trigger up to eleven different types of ADC captures.

The ADC Module has a dedicated “ADC Event Control” register (EVCTRL). The register controls the Compare Hit and Results Ready generator. To use generated events as a capture trigger **EVCTRL n .STARTEI** must be set. The polarity of trigger inputs can be inverted by setting **EVCTRL n .STARTINV**.

38.5.10 Power Management

The ADC will continue to operate in any sleep mode where the selected source clock is running. The ADC’s interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

Note: If several events are connected to the ADC, the enabled action will be taken on any of the incoming events.

38.5.11 Idle and Standby Sleep Mode Operation

The host device supports several modes of operation, starting at Active and ending with Off:

Mode of Operation:	Behavior:	Wake-Up Signal:
Active	CPU running. All memory is active. ADC can run.	None
(CPU) Idle	CPU halted. All memory is active. ADC ⁽¹⁾	All Interrupts
Standby	All logic content retained. ADC ⁽¹⁾ NVM in sleep mode. Configurable RAM retention support.	All Interrupts
Hibernate	Only backup domain remains active. ADC cannot run. Configurable RAM retention. Only 32 kHz clock can run.	Backup Interrupts
Backup	Only backup domain remains active. ADC cannot run. Only 32 kHz clock can run.	Backup Interrupts
Off	Everything is off.	External Reset

Note:

1. The CTRLA.ONDEMAND and CTRLA.RUNSTDBY bits control the behavior of the ADC during Idle and Standby sleep modes, in cases where the ADC is enabled (CTRLA.ENABLE = 1). See the following table:

Table 38-5. ADC Idle and Standby Mode Behavior

CTRLA.ENABLE	CTRLA.RUNSTDBY	CTRLA.ONDEMAND	Description
0	x	x	Disabled.
1	0	0	ADC inactive in Idle and Standby mode
1	0	1	Run in Idle and Standby modes only on EVSYS trigger request.
1	1	0	Run in Idle and Standby modes.
1	1	1	Run in Idle and Standby modes.

Note: When CTRLA.ONDEMAND=1, the analog block is always powered-off. When a start request is detected, the system returns from sleep and starts a new conversion. When RUNSTDBY is enabled the GCLK to the ADC continues to run when ONDEMAND=0. With ONDEMAND and RUNSTDBY set the GCLK will be requested. Therefore, there will be an additional delay until the analog front end is ready (CTLINTFLAG.CRRDY=1).

38.5.12 Debug Operation

Setting **DBGCTRL.DBGRUN** = 1 enables the ADC to continue operation when the device is halted in debug mode. If this bit is not set, then the ADC is halted in debug mode.

If **DBGCTRL.DBGRUN**=0, then during debug the ADC macro will capture all the triggers that fire and on exit from the debug operation to normal running operation the ADC will perform conversions on the captured event or events based on ADC channel priorities.

During a debug stop all the ADC registers are still write protected as long as **CTRLA.ENABLE**=1.

38.5.13 Register Synchronization

The **SYNCBUSY** register supports the **CTRLA** and **CTRLB** registers. When **CTRLA.ENABLE** has been set no additional writes to this bit are allowed as long as the SYNCYBUSY.ENABLE bit remains high. Additional writes to the **CTRLB** register are not allowed as long as the **SYNCBUSY.CTRLB** bit remains high.

Register synchronization is required for other registers in the GLCK clock domain. These registers can be modified while CTRLA.ENABLE=0, i.e. as long as the ADC is not enabled. Once the ADC is enabled (CTRLA.ENABLE=1) these registers are synchronized to the APB_CLK (Main Clock) domain and are write protected.

38.6 Peripheral Dependencies

Table 38-6. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
ADC	0x44818000	130 : REQ0 131 : REQ1	APB: CLKMSK[1] MASK17	PCHCTRL[33]	39 STATUS1 PERID7	37 : PFFRDY	<u>Generators:</u> CHRDY: CHANNEL[76] CMP: CHANNEL[77] <u>Users:</u> TRIG0: USER[38],Async TRIG1: USER[39],Async TRIG2: USER[40],Async TRIG3: USER[41],Async TRIG4: USER[42],Async TRIG5: USER[43],Async TRIG6: USER[44],Async TRIG7: USER[45],Async TRIG8: USER[46],Async TRIG9: USER[47],Async TRIG10: USER[48],Async	PD_CORE_SW

38.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0	ONDEMAND	RUNSTDBY		AIPMPEN		ANAEN	ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16								
		15:8	SWCNVEN				STRGEN	TRGSUSP	LSWTRG	GSWTRG
		7:0	SAMP	RQCNVRT	ADCORSEL[1:0]	ADCHSEL[3:0]				
0x08	CTRLC	31:24		COREINTERLEAVED[2:0]						
		23:16								
		15:8	CNT[15:8]							
		7:0	CNT[7:0]							
0x0C ... 0x0F	Reserved									
0x10	CTRLD	31:24		VREFSEL[2:0]			WKUPEXP[3:0]			
		23:16	ANLEN3	ANLEN2	ANLEN1	ANLEN0	CHNEN3	CHNEN2	CHNEN1	CHNEN0
		15:8			CTLCKDIV[5:0]					
		7:0								
0x14 ... 0x1F	Reserved									
0x20	CORCTRL	31:24		ADCDIV[6:0]						
		23:16		SCNRTDS	STRGLVL		STRGSRC[3:0]			
		15:8	EIRQOVR	EIS[2:0]			SELRES[1:0]		SAMC[9:8]	
		7:0	SAMC[7:0]							
0x24	CHNCFG1	31:24	LVL[15:8]							
		23:16	LVL[7:0]							
		15:8	CHNCMPEN[15:8]							
		7:0	CHNCMPEN[7:0]							
0x28	CHNCFG2	31:24	FRACT[15:8]							
		23:16	FRACT[7:0]							
		15:8	CSS[15:8]							
		7:0	CSS[7:0]							
0x2C	CHNCFG3	31:24	SIGN[15:8]							
		23:16	SIGN[7:0]							
		15:8	DIFF[15:8]							
		7:0	DIFF[7:0]							
0x30	CHNCFG4	31:24	TRGSRC7[3:0]				TRGSRC6[3:0]			
		23:16	TRGSRC5[3:0]				TRGSRC4[3:0]			
		15:8	TRGSRC3[3:0]				TRGSRC2[3:0]			
		7:0	TRGSRC1[3:0]				TRGSRC0[3:0]			
0x34	CHNCFG5	31:24	TRGSRC15[3:0]				TRGSRC14[3:0]			
		23:16	TRGSRC13[3:0]				TRGSRC12[3:0]			
		15:8	TRGSRC11[3:0]				TRGSRC10[3:0]			
		7:0	TRGSRC9[3:0]				TRGSRC8[3:0]			
0x38	CALCTRL	31:24	CALBITS[31:24]							
		23:16	CALBITS[23:16]							
		15:8	CALBITS[15:8]							
		7:0	CALBITS[7:0]							
0x3C	EVCTRL	31:24								
		23:16								
		15:8								
		7:0			CMPEO	RESRDYEO	STARTINV			STARTEI
0x40 ... 0xAF	Reserved									

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xB0	CMPCTRL	31:24			IEHIHI	IEHILO	ADCMPhi[11:8]			
		23:16	ADCMPhi[7:0]							
		15:8	IEBTWN	IELOHI	IELOLO	CMPEN	ADCMPL0[11:8]			
		7:0	ADCMPL0[7:0]							
0xB4 ... 0xBF	Reserved									
0xC0	FLTCTRL	31:24								
		23:16								
		15:8	FLTCHNID[3:0]							FLTEN
		7:0				DATA16EN	FMODE	OVRSAM[2:0]		
0xC4 ... 0xCF	Reserved									
0xD0	CORCHDATAID	31:24								
		23:16								
		15:8								
		7:0	CORDYID[1:0]			CHRDYID[3:0]				
0xD4	CHRDYDAT	31:24					FRACT	SIGN	DIFF	LVL
		23:16								
		15:8	CHRDYDAT[15:8]							
		7:0	CHRDYDAT[7:0]							
0xD8	PFFDATA	31:24	PFFCNT[7:0]							
		23:16	PFFFRACT	PFFSIGN	PFFCORID[1:0]			PFFCHNID[3:0]		
		15:8	PFFDATA[15:8]							
		7:0	PFFDATA[7:0]							
0xDC	DMABASE	31:24	DMABASE[31:24]							
		23:16	DMABASE[23:16]							
		15:8	DMABASE[15:8]							
		7:0	DMABASE[7:0]							
0xE0	DMACTRL	31:24								
		23:16								
		15:8						DMABL[2:0]		
		7:0	DMACR3	DMACR2	DMACR1	DMACR0		DMAEN		
0xE4	PFFCTRL	31:24								
		23:16								PFFRDYDM
		15:8								
		7:0	PFFCR3	PFFCR2	PFFCR1	PFFCR0		PFFEN		
0xE8	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0						CTRLB	ENABLE	SWRST
0xEC ... 0xEF	Reserved									
0xF0	DMAINTENCLR	31:24								
		23:16								SOVFL
		15:8								
		7:0	RBF[3:0]				RAF[3:0]			
0xF4	DMAINTSET	31:24								
		23:16								SOVFL
		15:8								
		7:0	RBF[3:0]				RAF[3:0]			
0xF8	DMAINTFLAG	31:24								
		23:16							DMAERR	SOVFL
		15:8	EBF[3:0]				EAF[3:0]			
		7:0	RBF[3:0]				RAF[3:0]			

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0xFC	CTLINTENSET	31:24								
		23:16								
		15:8					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
		7:0	VREFRDY	VREFUPD			CRRDY3	CRRDY2	CRRDY1	CRRDY0
0x0100	CTLINTENCLR	31:24								
		23:16								
		15:8					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
		7:0	VREFRDY	VREFUPD			CRRDY3	CRRDY2	CRRDY1	CRRDY0
0x0104	CTLINTFLAG	31:24								
		23:16								
		15:8					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
		7:0	VREFRDY	VREFUPD			CRRDY[3:0]			
0x0108 ... 0x011F	Reserved									
0x0120	INTENCLR	31:24	CHRDY[15:8]							
		23:16	CHRDY[7:0]							
		15:8					EOSRDY	CHNERRC	FLTRDY	CHRDYC
		7:0	SOVFL			CMPHIT				
0x0124	INTENSET	31:24	CHRDY[15:8]							
		23:16	CHRDY[7:0]							
		15:8					EOSRDY	CHNERRC	FLTRDY	CHRDYC
		7:0	SOVFL			CMPHIT				
0x0128	INTFLAG	31:24	CHRDY[15:8]							
		23:16	CHRDY[7:0]							
		15:8	CRDYID[3:0]				EOSRDY	CHNERRC	FLTRDY	CHRDYC
		7:0	SOVFL			CMPHIT	CMPINTID[3:0]			
0x012C ... 0x0167	Reserved									
0x0168	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								DBGRUN

38.7.1 CONTROL A REGISTER

Name: CTRLA
Offset: 0x000
Reset: 0x00000080
Property: RW

Table 38-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	ONDEMAND	RUNSTDBY		AIPMPEN		ANAEN	ENABLE	SWRST
Access	R/W	R/W		R/W		R/W	R/W	R/W
Reset	1	0		0		0	0	0

Bit 7 – ONDEMAND On Demand Control

The On Demand operation mode allows the ADC to be normally disabled, lowest power mode, until enabled by an EVSYS trigger event. The consequences of this however is that the analog logic is completely powered down when ONDEMAND=1 and clocks are suspended. Upon a EVSYS trigger event, it will require up to 20-50µs for the ADC to warm up meaning that the actual sample/convert sequence will not happen for:

ADC analog input signal sample point in time = (EVSYS Trigger event + CTRLD.WKUPEXP)

Which can be up to 50µs after the initial EVSYS trigger event. After the conversion is complete, if there are no pending EVSYS trigger events the ADC will again power down and the sequence will repeat as before.

When ONDEMAND = 1, the ADC works only on a rising edge EVSYS events. In addition, all other ADC trigger sources other than EVSYS triggers will not be acknowledged and therefore will render the ADC inactive. (See table below)

In Idle and Standby mode, if ONDEMAND = 1 and CTRLA.RUNSTDBY = 1, the ADC will be active and not powered down. If CTRLA.RUNSTDBY=0, the ADC is powered down and ADC clocks are suspended.

Note: CTRLD.WKUPEXP should be set by user to the equivalent of 50 µs or CTRLD.WKUPEXP[xxx] = 50 µs / TAD.

Table 38-8. ONDEMAND Matrix

SYSTEM STATUS	ONDEMAND	RUNSTDBY	ADC Trigger Source	ADC BEHAVIOR
ACTIVE	1	x	ADC Global Software Trigger.	Do Nothing
		x	ADC Global Level Software Trigger	Do Nothing
		x	ADC SCANTRG	Do Nothing except only when SCAN trigger source selected is EVSYS trigger, then RUN only when EVSYS trigger request is received.
		x	ADC STRIG Synchronous Trigger	Do Nothing
		x	ADC Trigger Event(s) from Event System (EVSYS)	RUN only when EVSYS trigger request is received.
	0	x	ADC Global Software Trigger.	ADC active, Trigger sample/conv and wait for next trigger
		x	ADC Global Level Software Trigger	ADC active, Trigger sample/conv continuously.
		x	ADC SCANTRG	ADC active, Trigger sample/conv and wait for next trigger
		x	ADC STRIG Synchronous Trigger	ADC active, Trigger sample/conv and wait for next trigger
		x	ADC Trigger Event(s) from Event System (EVSYS)	ADC active, Trigger sample/conv and wait for next trigger.
IDLE, STANDBY	0	1	All triggers working	ADC active
	1	0	n/a	Do nothing
	1	1	ADC Trigger Event(s) from Event System (EVSYS)	RUN only when EVSYS trigger request is received.

Value	Description
0	The ADC is always on, if ENABLE = 1. (Highly Recommended)
1	See table below.

Bit 6 – RUNSTDBY Run in Standby

This bit controls how the ADC behaves during Standby Sleep mode.

This bit is not synchronized.

Value	Description
0	Discontinue module operation when device enters idle/standby mode (but do not stop the CTL_CLK to preserve the analog biasing). The ADC will complete the ongoing conversion before disabling the module.
1	The ADC is not stopped in Idle or Standby sleep mode. If CTRLA.ONDEMAND = 1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND = 0, the ADC will always be running in standby sleep mode.

Bit 4 – AIPMPEN Charge Pump Enable

Bit 2 – ANAEN Analog Enable

Note: This bit must be set prior to setting any of the CTRLD.ANLENn bits.

Value	Description
0	The analog and bias circuitry for the ADC module is powered down and clocks disabled. Trigger events will not be serviced.
1	Enables and powers up the ADC module analog logic based on the settings of CTRLD.ANLEN0 = 1. Whenever the ADC module exits a power down state to a powered-up state, CTRLA.ANAEN = 1 and CTRLD.ANLEN0 = 1, then the system hardware will wait until CTRLD.WKUPEXP time has expired before starting an ADC sample/conversion if a trigger event is pending. The ADC required warm-up time is 20-50us before it will stabilize for a valid ADC sample/conversion sequence to start. CTRLA.ENABLE does not need to be set to power-up the analog logic and begin the CTRLD.WKUPEXP time out.

Bit 1 – ENABLE ENABLE BIT

Notes:

1. The ENABLE bit should be set only after the ADC module has been configured. Changing the configuration bits after enabling the ADC could result in unpredictable behavior.
2. When ENABLE = 0 the internal control logic is reset, and all status generated by the module is cleared. All ADC Module clocks are disabled unless ANAEN = 1. In this case ANLEN0 = 0 disable clocks and ANLEN0 = 1 enables them. All ADC registers are available for reading and writing.

Value	Description
0	A/D converter is off.
1	A/D converter module is enabled.

Bit 0 – SWRST SOFTWARE RESET BIT

Writing a one to the SWRST bit resets the state of the ADC and all the registers to their initial state. The only exception is the DBGCTRL.DBGRUN bit, which will keep its value after a SWRST. The module will be disabled after the reset (ENABLE = 0). When writing a one to SWRST, no other bits in the same register will be written, as SWRST will clear all the bits in the same register.

After writing a one to SWRST, SWRST will read back one until the module and the registers are reset. Any register write access during the ongoing reset will be discarded and an error will be generated. Read access can be performed without an error generated and must return reset value. Writing a one to SWRST will have priority above all other actions and will always happen immediately.

Writing a zero to SWRST has no effect.

Note: During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

38.7.2 CONTROL B REGISTER

Name: CTRLB
Offset: 0x004
Reset: 0x00000000
Property: RW

Table 38-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	SWCNVEN				STRGEN	TRGSUSP	LSWTRG	GSWTRG
Access	R/W				R/W	R/W	R/W	R/W
Reset	0				0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SAMP	RQCNVRT	ADCORSEL[1:0]		ADCHSEL[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – SWCNVEN Software Conversion Enable

Note: When CTRLB.SWCNVEN=1, the channels sample and conversions are controlled exclusively by SAMP and RQCNVRT. All hardware triggers and GSWTRG and LSWTRG are disabled. It will place the ADC module and all of its channels under the exclusive control of the CTRLB.SAMP and CTRLB.RQCNVRT bits.

Value	Description
0	Software controlled conversions disable. Traditional ADC hardware triggers as defined by CHNCFG4 and CHNCFG5 will be active if enabled.
1	SAMP and RQCNVRT bits control entire ADC module sample and convert respectively for all channels. Setting this bit blocks all other hardware triggers events defined by CHNCFG4 and CHNCFG5.

Bit 11 – STRGEN Synchronous Trigger Enable

Note: This bit is ignored if CTRLB.SWCNVEN=1.

Value	Description
0	Disable automatic ADC hardware CTRLC.CNT counter driven synchronous ADC triggers.
1	Enable automated hardware synchronous trigger period defined by CTRLC.CNT register.

Bit 10 – TRGSUSP Trigger Suspend

Value	Description
0	Trigger suspend disabled. Triggers if enabled and selected will occur.
1	Blocks triggers from starting new ADC conversions but does not disable the ADC Modules or disable trigger capture, (i.e. persistent last trigger event is latched). If trigger capture during trigger suspend is not desired for any channels, then CTRLD.CHNEN0 must be cleared then set in that order prior to resetting TRGSUSP. Pulsing down and up CTRLD.CHNEN0 prior to resetting TRGSUSP will clear ALL pending triggers.

Bit 9 – LSWTRG Level Global Trigger

Notes:

1. This bit is ignored if SWCNVEN=1.
2. If SWCNVEN=0, this bit is NOT self-clearing and is meant to allow the user software to implement continuous sample/conversions on the associated analog input channel.

Value	Description
0	Global Level Software Trigger disabled.
1	Trigger A/D conversions for ADC analog input(s) "y" that have selected LSWTRG bit as the trigger signal via the associated CHNCFG4.TRGSRC[y] or CHNCFG5.TRGSRC[y] = 0b0010 or via the CORCTRL.STRGSRC.

Bit 8 – GSWTRG Global Software Trigger

This software settable bit will trigger ADC sample/conversion sequences for ADC inputs that have selected the GSWTRG bit as the trigger signal via the associated CHNCFG4.TRGSRC[y] or CHNCFG5.TRGSRC[y], via the CORCTRL.STRGSRC value. This bit is auto cleared on the next APB clock cycle and is meant to implement single conversions on trigger edge-sensitive channels.

Note:

1. This bit is ignored if SWCNVEN=1.

Value	Description
0	Disable Global Software Trigger.
1	If SWCNVEN=0, this bit is auto cleared by hardware after sampling cycle has been triggered and is meant to implement single sample/conversions on trigger edge-sensitive channels.

Bit 7 – SAMP Channel Sample

Notes:

1. This bit is ignored if SWCNVEN = 0.
2. ADCHSEL[3:0] must be initialized at or before when the SAMP bit is set.
3. The SAMP bit will keep the S&H circuit in Sample mode until the bit is cleared by the users software. Also, usage of the SAMP bit will cause settings of respective CORCTRL.SAMC<9:0> bits to be ignored.
4. The SAMP bit is not a self-clearing bit and it is the responsibility of application software to clear this bit but only after setting the RQCNVRT bit to start the analog-to-digital conversion.
5. When the SAMP and RQCNVRT bits are used by software routines, all TRGSRCx<4:0> bits and STRGSRC<4:0> bits should be set to '00000' to disable all hardware triggers and prevent them from interfering with the software-controlled sampling command signal SAMP and with the software controlled trigger RQCNVRT.
6. The SAMP bit should only be used in conjunction with RQCNVRT for user software-controlled sampling and triggering.

Value	Description
0	ADC is not software-controlled. Sampling the channel selected by ADCHSEL[3:0].
1	ADC is sampling and remain sampling for as long as user has this bit set in SW. The sampled analog input is defined by ADCHSEL[3:0] provided SWCNVEN = 1.

Bit 6 – RQCNVRT Request Channel Convert

Notes:

1. If user's software sets RQCNVRT = 1, an ADC conversion will begin immediately and terminate the sampling period defined by CORCTRL.SAMC. This bit is cleared by hardware after the conversion is complete.
2. If SWCNVEN = 1, after setting this bit the users software must immediately clear the SAMP bit.
3. This bit is ignored if SWCNVEN = 0, (i.e. if software-controlled sample/conversions are disabled).

Value	Description
0	ADC is not converting if SWCNVEN = 1 or user previously set this bit and the previous ADC conversion is complete for the analog channel that was defined by ADCHSEL[3:0]
1	Terminate sampling and begin conversion of ADC and analog channel defined by ADCHSEL[3:0]. This bit is cleared by hardware when the conversion is complete.

Bits 5:4 – ADCORSEL[1:0] Software Trigger Core Select

Bits 3:0 – ADCHSEL[3:0] Software Trigger Channel Select

This binary encoded bit field selects the ADC analog input to be sampled and converted respectively by the SAMP and RQCNVRT bit if SWCNVEN = 1.

Note: These bits are ignored if SWCNVEN = 0.

ADCHSEL[3:0]	Analog Channel Input
0x0	AIN0
0x1	AIN1
0x2	AIN2
0x3	AIN3
0x4	AIN4
0x5	AIN5
0x6	AIN6
0x7	AIN7
0x8	AIN8
0x9	AIN9
0xA	AIN10
0xB	AIN11
0xC	VDDCORE_SW
0xD	Reserved
0xE	1.2V IVREF
0xF	Reserved

38.7.3 Control C Register

Name: CTRLC
Offset: 0x008
Reset: 0x00000000
Property: RW

Table 38-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		COREINTERLEAVED[2:0]						
Access		R/W	R/W	R/W				
Reset		0	0	0				
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	CNT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30:28 – COREINTERLEAVED[2:0] Number of Core to Interleave Triggers

Bits 15:0 – CNT[15:0] Clock Divider for Synchronous Trigger

Free-running counter based on CTL_CLK times out when it reaches this value. At time out, the STRIG synchronous trigger will fire.

Notes:

1. This register is not valid unless either [CORCTRLn.STRGSRC=0x4 plus CHNCFG40/50.TRGSRCx=0x3 plus CHNCFG20.CSSx=1] or [CHNCFG40/50.TRGSRCx=0x4 plus CTRLB.SWCNVEN=0] for Synchronous Trigger from CTRLC.CNT.
2. $CTL\_CLK = GCLK\_ADC / (CTRLD.CTLCKDIV+1)$
3. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1 and return a bus error).

38.7.4 Control D Register

Name: CTRLD
Offset: 0x010
Reset: 0x00000000
Property: RW

Table 38-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		VREFSEL[2:0]			WKUPEXP[3:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ANLEN3	ANLEN2	ANLEN1	ANLEN0	CHNEN3	CHNEN2	CHNEN1	CHNEN0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
			CTLCKDIV[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 30:28 – VREFSEL[2:0] Voltage Reference Select

VREFSEL[2:0]	ADREF+	ADREF-
000	A_{VDD}	A_{VSS}
001	External V_{REFH}	A_{VSS}
010–111	Reserved	Reserved

Note:

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

Value	Name	Description
0	AVDD_AVSS	AVDD and AVSS
1	EXTERNAL_VREFH_AVSS	External VREFH and AVSS

Bits 27:24 – WKUPEXP[3:0] Wakeup cycles

These bits represent the number of ADC TAD clocks required to warm-up the ADC module before it can perform conversion.

1111 = 215 = 32,768 TAD clocks

0110 = 26 = 64 TAD clocks

0101 = 25 = 32 TAD clocks
0100 = 24 = 16 TAD clocks
0011 = 24 = 16 TAD clocks
0010 = 24 = 16 TAD clocks
0001 = 24 = 16 TAD clocks
0000 = 24 = 16 TAD clocks

Notes:

1. Minimum required ADC warm-up time, (i.e., WKUPCLKCNT), is 50 μ s. (i.e., TAD * WKUPCLKCNT).
2. $TAD = [1 / ((GCLK\_ADC \text{ Freq} / (CTRLD.CTLCKDIV+1)) / (2 * CORCTRL.ADCDIV))] \text{ or } (GCLK\_ADC \text{ Period} * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV).$
3. These bits are Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

Bit 23 – ANLEN3 Analog and Bias circuitry Enable for core 3

Bit 22 – ANLEN2 Analog and Bias circuitry Enable for core 2

Bit 21 – ANLEN1 Analog and Bias circuitry Enable for core 1

Bit 20 – ANLEN0 Analog and Bias circuitry Enable for core 0

Notes:

1. The CALCTRL register must be initialized by the users software to the factory-provided values in the Calibration Configuration Register FCCFG65 before setting ANLEN0=1.
2. CTRLA.ANAEN=1 must be set prior to setting ANLEN0=1.
3. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

Value	Description
0	ADC Analog and bias circuitry powered down and clocks suspended. Analog logic power saving mode.
1	Analog and bias circuitry enabled. Once the analog and bias circuit is enabled CTRLA.ANLEN=1 and ANLEN, the ADC module needs a warm-up time, as defined by the WKUPEXP<3:0> bits.

Bit 19 – CHNEN3 Core 3 Digital Channel Enable

Bit 18 – CHNEN2 Core 2 Digital Channel Enable

Bit 17 – CHNEN1 Core 1 Digital Channel Enable

Bit 16 – CHNEN0 Core 0 Digital Channel Enable

Notes:

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.
2. ANLEN0 qualifies CHNEN0: If ANLEN0 =0, then digital logic is also disabled.

Value	Description
0	ADC digital logic disabled. No trigger, sample or conversion events will be processed. (power-saving mode with fast 2 TAD clock wakeup provided ANLEN0 = 1).
1	ADC digital logic enabled (required for active operation).

Bits 13:8 – CTLCKDIV[5:0] Control Clock Divider

The CTLCKDIV bit field divides the GCLK_ADC input clock into the ADC Module control clock CTL_CLK scaled by CTRLD.CTLCKDIV with period T_Q .

Note: This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and returns a bus error.

Value	Description
111111	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 64 \cdot T_{GCLK\_ADCx} = T_Q$
111110	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 63 \cdot T_{GCLK\_ADCx} = T_Q$
...	...
000100	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 5 \cdot T_{GCLK\_ADCx} = T_Q$
000011	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 4 \cdot T_{GCLK\_ADCx} = T_Q$
000010	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 3 \cdot T_{GCLK\_ADCx} = T_Q$
000001	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 2 \cdot T_{GCLK\_ADCx} = T_Q$
000000	$T_{GCLK\_ADCx} \cdot (CTLCKDIV[5:0] + 1) = 1 \cdot T_{GCLK\_ADCx} = T_Q$

38.7.5 SARCORE Control

Name: CORCTRL
Offset: 0x020
Reset: 0x00000C00
Property: RW

Table 38-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
		SCNRTDS	STRGLVL					
Access		R/W	R/W		R/W	R/W	R/W	R/W
Reset		0	0		0	0	0	0
Bit	15	14	13	12	11	10	9	8
	EIRQOVR		EIS[2:0]			SELRES[1:0]		SAMC[9:8]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0
Bit	7	6	5	4	3	2	1	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 30:24 – ADCDIV[6:0] Division Ratio for SARCORE clock

ADCDIV provides the ADC analog functional clock referred to as the ADC clock or TAD, (i.e., the period of the ADC clock). It divides CTL_CLK output clock from (GCLK_ADC Period * (CTRLD.CTLCKDIV+1)) with period = TQ to produce the resulting ADC_CLK for ADC sampling and converting. It will have the period TAD given by the formula:

$$TAD = (GCLK\_ADC \text{ Period} * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV)$$

Note: The ADC throughput rate $F_{TPR} = [1 / ((SAMC+2) * TAD) + ((\#bits \text{ of resolution selected}+1)*TAD)] / \# \text{ of active AINx inputs.}$

Example 1:

$TAD = 1/ADC\_CLK = 1/75 \text{ MHz} = 13.33 \text{ ns}$, $SAMC = 0x1$, $SELRES = 0x3 = 12 \text{ bits}$, AIN0 used only ADC throughput rate

$$F_{TPR} = [1 / (3TAD + 13TAD)] / 1$$

$$= 1 / 16TAD$$

$$= 1 / (16 * 13.33e-9)$$

$$= 4.6875 \text{ msp}$$

Example 2: (Non Interleaved Mode)

$TAD = 1/ADC\_CLK = 1/75 \text{ MHz} = 13.33 \text{ ns}$, $SAMC = 0x1$, $SELRES = 0x3 = 12 \text{ bits}$, AIN0 and AIN3 in single ended mode used ADC throughput rate

$$F_{TPR} = [1 / (3TAD + 13TAD)] / 2$$

$$= (1 / 16TAD) / 2$$

$$= (1 / (16 * 13.33e-9)) / 2$$

$$= 4.6875e+6$$

$$= 2.34375 \text{ msp}$$

Notes:

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1) . Returns a bus error.
2. The minimum sample rate for ADC0 is $F_{\text{pr}} \geq 100 \text{ ksp}$.
3. For optimal performance and accuracy, the user must select the maximum TAD clock supported by the device as specified in the ADC electrical specs:

$$\text{TAD} = (\text{GCLK\_ADCx Period} * (\text{CTRLD.CTLCKDIV}+1)) * (2 * \text{ADCDIV})$$
4. If the user wishes to control the ADC throughput rate, they must do so by modulating either CORCTRLn.SAMC or the EVSYS trigger interval or both. Do not try to do so by reducing the TAD clock frequency as this will increase the ADC conversion time allowing the charge on the ADC holding capacitor from the sampled analog input signal to leak off, attenuating the ADC result accuracy.

Value	Description
11111111	$254 \cdot T_Q = T_{\text{AD}}$
...	...
0000011	$6 \cdot T_Q = T_{\text{AD}}$
0000010	$4 \cdot T_Q = T_{\text{AD}}$
0000001	$2 \cdot T_Q = T_{\text{AD}}$
0000000	Reserved

Bit 22 – SCNRDTS SCAN Re-trigger Disable

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

Value	Description
0	Allows the scan cycle to restart from the beginning, lowest CSSn channel even before all of the scan channels have been measured if a new scan trigger arrives before then. If this happens then the INTENCLR.EOSRDY flag will be set when the current scan is interrupted, and the new scan starts from the beginning of the lowest CSSn channel selected.
1	Prevents an early scan trigger (arriving before of the end of the current scan) from starting a new scan cycle. The scan will include all channels which have their CHNCFG4.TRGSRCk set to point to the Scan Trigger, i.e., CHNCFG4.TRGSRCy[3:0] = 4'b0011, and have their associated CHNCFG2.CSSk bit set.

Bit 21 – STRGLVL Scan Trigger Level Sensitivity

Note:

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

STRGLVL functions as follows:

Value	Description
0	SCANTRG is Positive Edge Active(the power-up value for backwards compatibility). A positive edge on the SCANTRG will initiate a single but complete scan of all included channels.
1	SCANTRG is High Level Active. So long as SCANTRG stays high, the entire scan will re-trigger.

Bits 19:16 – STRGSRC[3:0] SCAN trigger source selection

These bits select the trigger source for the scan trigger SCANTRG. The trigger STRIG serves all the channels k, which have their CHNCFG{4|5}.TRGSRCk = 4'h0011, and have their Channel Scan Select bit set (CHNCFG2.CSSk = 1'b1).

Notes:

1. This bit is Enabled Protected: Writes are ignored when CTRLA.ENABLE = 1 returns a bus error.
2. In order to utilize CORCTRLn.STRGSRC=0x4, user must configure the following registers:
 - a. CTRLC.CNT.
 - b. CHNCFG2n.CSSx.
 - c. CHNCFG4/5n.TRGSRCx=0x3.
 - d. CTRLB.SWCNVEN = 0.

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bit 15 – EIRQOVR Interrupt Type Select

Note:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) . returns a bus error.

Value	Description
0	Use normal ADC interrupts
1	Use early ADC interrupts as defined by CORCTRL.EIS bits

Bits 14:12 – EIS[2:0] Early Interrupt Select

These bits select the number of core clocks and TAD clocks prior to the end of conversion at which the early interrupt is generated. All channels share the same EIS setting.

The interrupt is generated ((EIS + 1) x TAD) ADC Module clocks prior to end of conversion.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) . returns a bus error.
2. Early interrupt is a feature that can be useful in ADC very high speed data acquisitions to mask the MCU interrupt latency delay.

Example: CPU MCLK = 72 Mhz, ADC CTL_CLK = 36 MHz and ADC TAD Clk = 18 MHz.

Hypothetically, if it took the CPU 20 MCLK cycles to service an interrupt, then theoretically the ADC early interrupt EIS could be set to 4, hence the ISR latency was masked. The CPU could then read the ADC result almost immediately from within the Interrupt Service Routine (ISR), when the result was ready without any concern for overwriting the current result or FIFO with the next ADC conversion result.

Notes: Depending on the bit resolution selection field SELRES, the allowed maximum EIS values are: as follows

- 12-bit Resolution, all 8 possible settings are allowed, 0 - 7
- 10-bit Resolution, all 8 possible settings are allowed, 0 - 7
- 8-bit Resolution, only the 6 lowest settings are allowed, 0 - 5
- 6-bit Resolution, only the 4 lowest settings are allowed, 0 - 3

The hardware will utilize the maximum allowed EIS setting if the user sets the EIS value that is too big.



Do not set the EIS to large or the interrupt can occur before the ADC conversion is complete and the CPU could read the previous ADC conversion result instead of the one in progress.

Value	Description
0x0	Generate interrupt 1 TAD clock before end of ADC conversion
0x1	Generate interrupt 2 TAD clocks before end of ADC conversion
0x2	Generate interrupt 3 TAD clocks before end of ADC conversion
0x3	Generate interrupt 4 TAD clocks before end of ADC conversion
0x4	Generate interrupt 5 TAD clocks before end of ADC conversion
0x5	Generate interrupt 6 TAD clocks before end of ADC conversion
0x6	Generate interrupt 7 TAD clocks before end of ADC conversion
0x7	Generate interrupt 8 TAD clocks before end of ADC conversion

Bits 11:10 – SELRES[1:0] Selects Resolution

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

Value	Name	Description
0	6_BITS	6 bits
1	8_BITS	8 bits
2	10_BITS	10 bits
3	12_BITS	12 bits (power-on default)

Bits 9:0 – SAMC[9:0] Sample Count

The sample time required depends on the external analog signal source impedance. (See ADC electrical characteristics of SAMC values required based on external source impedance. If the external analog source impedance is unknown or if your getting inconsistent ADC results consider increasing the SAMC sample time at the cost of a lower ADC throughput rate of course.

ADC Throughput Rate

$$F_{TPR} = [1 / ((SAMC \text{ value} + 2) * TAD) + ((\# \text{ bits of resolution selected} + 1) * TAD)] / \# \text{ of active AINx inputs}$$

Notes:

1. All channels share the same SAMC setting.
2. The sampling sequence starts with a trigger event.
3. The internal sampling cap is not discharged between samples.
4. These bits are Enabled Protected: Writes are ignored when CTRLA.ENABLE = 1 returns a bus error.
5. When IVREF is to be sampled (SAMC value + 2) * TAD must be $\geq 30 \mu s$.

Value	Description
11111111	1025 T_{ADn}
1	

Value	Description
...	...
000000000 1	3 T_{ADn}
000000000 0	Reserved

38.7.6 Channel Configuration 1 (LVL/CMPEN)

Name: CHNCFG1
Offset: 0x024
Reset: 0x00000000
Property: RW

Table 38-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	LVL[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	LVL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CHNCMPEN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHNCMPEN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – LVL[15:0] Channel Level

Bits 15:0 – CHNCMPEN[15:0] Channel Comparator Enable

38.7.7 Channel Configuration 2(FRACT/CSS)

Name: CHNCFG2
Offset: 0x028
Reset: 0x00000000
Property: RW

Table 38-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FRACT[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FRACT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CSS[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CSS[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – FRACT[15:0] Channel Fractional

Bits 15:0 – CSS[15:0] Channel SCAN Select

38.7.8 Channel Configuration3 (SIGN/DIFF)

Name: CHNCFG3
Offset: 0x02C
Reset: 0x00000000
Property: RW

Table 38-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SIGN[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	SIGN[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DIFF[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DIFF[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – SIGN[15:0] SIGN setting

Bits 15:0 – DIFF[15:0] Differential Mode

38.7.9 Channel Configuration 4 (TRGSRC)

Name: CHNCFG4
Offset: 0x030
Reset: 0x00000000
Property: RW

Table 38-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRGSRC7[3:0]				TRGSRC6[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TRGSRC5[3:0]				TRGSRC4[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TRGSRC3[3:0]				TRGSRC2[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TRGSRC1[3:0]				TRGSRC0[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – TRGSRC7[3:0] Channel 7 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 27:24 – TRGSRC6[3:0] Channel 6 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 23:20 – TRGSRC5[3:0] Channel 5 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 19:16 – TRGSRC4[3:0] Channel 4 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9

Value	Name	Description
15	EVENT_USER10	ADC Trigger Event User 10

Bits 15:12 – TRGSRC3[3:0] Channel 3 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 11:8 – TRGSRC2[3:0] Channel 2 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 7:4 – TRGSRC1[3:0] Channel 1 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4

Value	Name	Description
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 3:0 – TRGSRC0[3:0] Channel 0 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

38.7.10 Channel Configuration 5 (TRGSRC)

Name: CHNCFG5
Offset: 0x034
Reset: 0x00000000
Property: RW

Table 38-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TRGSRC15[3:0]				TRGSRC14[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TRGSRC13[3:0]				TRGSRC12[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TRGSRC11[3:0]				TRGSRC10[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TRGSRC9[3:0]				TRGSRC8[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – TRGSRC15[3:0] Channel 15 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 27:24 – TRGSRC14[3:0] Channel 14 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 23:20 – TRGSRC13[3:0] Channel 13 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 19:16 – TRGSRC12[3:0] Channel 12 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9

Value	Name	Description
15	EVENT_USER10	ADC Trigger Event User 10

Bits 15:12 – TRGSRC11[3:0] Channel 11 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 11:8 – TRGSRC10[3:0] Channel 10 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 7:4 – TRGSRC9[3:0] Channel 9 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4

Value	Name	Description
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

Bits 3:0 – TRGSRC8[3:0] Channel 8 Trigger Source

Value	Name	Description
0	NO_TRIGGER	No Trigger (NOP)
1	GLOBAL_SOFTWARE_TRIGGER	Global Software Trigger
2	GLOBAL_LEVEL_TRIGGER	Global Level Software Trigger
3	SCAN_TRIGGER	SCANTRG - Scan Trigger
4	SYNC_TRIGGER	Synchronous Trigger (STRIG)
5	EVENT_USER0	ADC Trigger Event User 0
6	EVENT_USER1	ADC Trigger Event User 1
7	EVENT_USER2	ADC Trigger Event User 2
8	EVENT_USER3	ADC Trigger Event User 3
9	EVENT_USER4	ADC Trigger Event User 4
10	EVENT_USER5	ADC Trigger Event User 5
11	EVENT_USER6	ADC Trigger Event User 6
12	EVENT_USER7	ADC Trigger Event User 7
13	EVENT_USER8	ADC Trigger Event User 8
14	EVENT_USER9	ADC Trigger Event User 9
15	EVENT_USER10	ADC Trigger Event User 10

38.7.11 SARCORE Calibration Value

Name: CALCTRL
Offset: 0x038
Reset: 0x00000000
Property: RW

Table 38-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CALBITS[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CALBITS[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CALBITS[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CALBITS[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CALBITS[31:0] Calibration Values

These register bits can only change when the applicable CTRLD.ANLEN0 bit is 0. Once the applicable CTRLD.ANLENN bit rising edge occurs, those bit values must remain unchanged until after the same CTRLD.ANLEN0 bit falling edge occurs.

This register must be initialized in user software to the factory-provided values in the Calibration Configuration Register FCCFG65 before setting CTRLD.ANLENN to 1.

Note:

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

38.7.12 Event Control

Name: EVCTRL
Offset: 0x03C
Reset: 0x00000000
Property: RW

Table 38-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			CMPEO	RESRDYEO	STARTINV			STARTEI
Access			R/W	R/W	R/W			R/W
Reset			0	0	0			0

Bit 5 – CMPEO Comparator Window Event Out

This bit indicates whether the Digital Comparator Window event output is enabled or not and whether an output event will be generated when the Digital Comparator Window is met.

Note: This event corresponds to EVSYS Event Generator 77.

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	Window event output is disabled, and an event will not be generated.
1	Window event output is enabled, and an event will be generated.

Bit 4 – RESRDYEO Result Ready Event Out

This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

Note: This event correspond to EVSYS Event Generator 76.

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	Result Ready event output is disabled, and an event will not be generated.

Value	Description
1	Result Ready event output is enabled, and an event will be generated.

Bit 3 – STARTINV Start Conversion Invert

Notes:

1. Only rising edge EVSYS ADC Trigger events are acknowledged.
2. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	Start event input source is not inverted.
1	Start event input source is inverted.

Bit 0 – STARTEI Start Event conversion input enable

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Value	Description
0	Not enabled. ADC Event System events cannot trigger start of conversions.
1	Enabled. ADC Event System events can trigger start of conversions.

38.7.13 Comparator Control

Name: CMPCTRL
Offset: 0x0B0
Reset: 0x00000000
Property: RW

Table 38-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
			IEHIHI	IEHILO	ADCM PHI[11:8]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	ADCM PHI[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	IEBTWN	IELOHI	IELOLO	CMPEN	ADCM PLO[11:8]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ADCM PLO[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 29 – IEHIHI Enable VAL ≥ CMPHI

Setting this bit enables comparison events $ADCM PHI \leq ADC$ value.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefore in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Bit 28 – IEHILO Enable VAL < CMPHI

Setting this bit enables comparison events $ADC \text{ value} < ADCM PHI$.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Bits 27:16 – ADCMPHI[11:0] High Limit of Digital Comparator

This register stores the limit value which is used for comparisons with the ADC Module output data when

IEHIHI = 1, IEHILO = 1, or IEBTWN = 1.

The user is responsible for formatting the data in ADCMPHI[11:0] as signed or unsigned to match the data format as specified by the CHNCFG3.SIGNk and CHNCFG2.FRACTk bits for all the analog input channels k which are enabled by CHNCFG1.CHNCMPENk .

Note: In Filter accumulation mode, the comparison is done on the upper 12 of the 16 bits of filter data.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Bit 15 – IEBTWN Enable CMPLO <= VAL < CMPHI

Setting this bit enables comparison events $ADCMPLO \leq ADC \text{ Value} < ADCMPHI$.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Bit 14 – IELOHI Enable VAL >= CMPLO

Setting this bit enables comparison events $ADCMPLO \leq ADC \text{ Value}$.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Value	Description
0	Use normal interrupts
1	use early interrupts

Bit 13 – IELOLO Enable VAL < CMPLO

Setting this bit enables comparison events ADC Value < ADCMPLO.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Bit 12 – CMPEN Comparator Enable

Setting this bit enables digital comparisons for the inputs to the ADC.

For each channel input channel k to be compared the corresponding bit CHNCFG1.CHNCMPENk must be set for the channel to be monitored.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

Bits 11:0 – ADCMPLO[11:0] Low Limit of Digital Comparator

This register stores the limit value which is used for comparisons with the ADC Module output data when

IELOHI = 1, IELOLO = 1, or IEBTWN = 1.

The user is responsible for formatting the data in ADCMPLO[11:0] as signed or unsigned to match the data format as specified by the CHNCFG3.SIGNk and CHNCFG2.FRACTk bits for all the analog input channels k which are enabled by CHNCFG1.CHNCMPENk .

Note: In Filter accumulation mode, the comparison is done on the upper 12 of the 16 bits of filter data.

Notes:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

38.7.14 Filter Control

Name: FLTCTRL
Offset: 0x0C0
Reset: 0x00000000
Property: RW

Table 38-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 13:10 – FLTCHNID[3:0] Channel ID

Identifies which input channel, k , is to be filtered by the Digital Filter.

Note:

1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Bit 8 – FLTEN Filter Enable

When set, this bit enables the Digital Filter to filter the output data generated by the ADC. The input channel to be filtered is determined by FLTCHNID.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Bit 4 – DATA16EN 16bit Averaging Mode

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

This bit is significant only if FMODE = 1 (Averaging Mode) and CHNCFG2.FRACTk= 1 (Fractional Output Mode, where k = FLTCHNID[3:0] is the chosen input for filtering) as follows:

Value	Description
0	Only the first 12 bits are significant, followed by 4 zeros.
1	All 16 bits of the filter output data are significant

Bit 3 – FMODE Filter Mode

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Value	Description
0	Filtering in Oversampling Mode (power-up default)
1	Filtering in Averaging Mode

Bits 2:0 – OVRSAM[2:0] Oversampling Ratio

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Determines the number of samples generated in the burst mode used for computing one single filter output value.

The OVRSAM encoding depends on the FMODE setting as follows:

If FMODE = 1 (Oversampling Mode) then OVRSAM is encoded as follows:

Value	Description
000	4 samples, shift sum 1-bit to right, output data is 13-bits
001	16 samples, shift sum 2-bits to right, output data is 14-bits
010	64 samples, shift sum 3-bits to right, output data is 15-bits
011	256 samples, shift sum 4-bits to right, output data is 16-bits
100	2 samples, shift sum 0-bits to right, output data is in 12.1 format
101	8 samples, shift sum 1-bit to right, output data is in 13.1 format
110	32 samples, shift sum 2-bits to right, output data is in 14.1 format
111	128 samples, shift sum 3-bits to right, output data is in 15.1 format

If FMODE=0 (Averaging Mode), then OVRSAM is encoded as follows:

Value	Description
000	2 samples to be averaged
001	4 samples to be averaged
010	8 samples to be averaged
011	16 samples to be averaged
100	32 samples to be averaged
101	64 samples to be averaged
110	128 samples to be averaged
111	256 samples to be averaged

Value	Name	Description
0	4_SAMPLES	(If FMODE is 0) 4 samples (shift sum 1 bit to right, output data is 13-bits) / (If FMODE is 1) 2 samples to be averaged
1	16_SAMPLES	(If FMODE is 0) 16 samples (shift sum 2 bits to right, output data is 14-bits) / (If FMODE is 1) 4 samples to be averaged
2	64_SAMPLES	(If FMODE is 0) 64 samples (shift sum 3 bits to right, output data is 15-bits) / (If FMODE is 1) 8 samples to be averaged
3	256_SAMPLES	(If FMODE is 0) 256 samples (shift sum 4 bits to right, output data is 16-bits) / (If FMODE is 1) 16 samples to be averaged
4	2_SAMPLES	(If FMODE is 0) 2 samples (shift sum 0 bits to right, output data is 12.1 format) / (If FMODE is 1) 32 samples to be averaged
5	8_SAMPLES	(If FMODE is 0) 8 samples (shift sum 1 bit to right, output data is 13.1 format) / (If FMODE is 1) 64 samples to be averaged
6	32_SAMPLES	(If FMODE is 0) 32 samples (shift sum 2 bits to right, output data is 14.1 format) / (If FMODE is 1) 128 samples to be averaged
7	128_SAMPLES	(If FMODE is 0) 128 samples (shift sum 3 bits to right, output data is 15.1 format) / (If FMODE is 1) 256 samples to be averaged

38.7.15 Channel Ready DATA ID

Name: CORCHDATAID
Offset: 0x0D0
Reset: 0x00000000
Property: RW

Table 38-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			CORDYID[1:0]		CHRDYID[3:0]			
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 5:4 – CORDYID[1:0] Core Read ID

Bits 3:0 – CHRDYID[3:0] Channel Read ID

Input Channel Index k:

For ADC, set value to input channel index, for status register CHNRDYDAT to display the current values of configuration bits and the last converted output data or written by the user to display the channel.

Note:

1. Selecting unimplemented input channels on a given ADC will return a bus error with the data (32'h00000000).

Value	Description
1110	analog input channel 14
...	...
0010	analog input channel 2
0001	analog channel 1
0000	analog channel 0

38.7.16 Channel Ready Data Register

Name: CHRDYDAT
Offset: 0x0D4
Reset: 0x00000000
Property: RW

Note: All the data read in this register pertains to the Module defined by CORCHDATAID.CORDYID and its analog input channel defined by CORCHDATAID.CHRDYID.

Table 38-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					FRACT	SIGN	DIFF	LVL
Access					R/K	R/K	R/K	R/K
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	CHRDYDAT[15:8]							
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CHRDYDAT[7:0]							
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0

Bit 27 – FRACT Fractional Setting

Bit 26 – SIGN Sign Setting

Bit 25 – DIFF Differential Setting

Bit 24 – LVL Level Setting

Bits 15:0 – CHRDYDAT[15:0] Channel Output Data

Note: A read of CHRDYDAT will generate a read bus error on analog input channels which have not been implemented and the returned data will be 0x0000_0000.

38.7.17 APB FIFO Output Data

Name: PFFDATA
Offset: 0x0D8
Reset: 0x00000000
Property: R

Note: Reading any part of the PFFDATA register advances the FIFO pointer, therefore it is recommended to read all 32 bits of the register and then parse out the values of each field.

Table 38-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PFFCNT[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PFFFRAC	PFFSIGN	PFFCORID[1:0]		PFFCHNID[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PFFDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PFFDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – PFFCNT[7:0] Current Data Entries in APB FIFO

Bit 23 – PFFFRAC Fractional Setting from APB FIFO

Bit 22 – PFFSIGN Channel Sign from the APB FIFO

Bits 21:20 – PFFCORID[1:0] Core ID from APB FIFO

Bits 19:16 – PFFCHNID[3:0] Channel ID from APB FIFO

Bits 15:0 – PFFDATA[15:0] SARCORE Conversion data from the APB FIFO
Data is in the format given by PFFFRAC and PFFSIGN.

38.7.18 DMA Sample Base Address

Name: DMABASE
Offset: 0x0DC
Reset: 0x00000000
Property: RW

Table 38-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DMABASE[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DMABASE[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DMABASE[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DMABASE[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DMABASE[31:0] DMA Sample Value Base Address

38.7.19 DMA Control Register

Name: DMACTRL
Offset: 0x0E0
Reset: 0x00000000
Property: RW

Table 38-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							DMABL[2:0]	
Access						R/W	R/W	R/W
Reset						0	0	0
Bit	7	6	5	4	3	2	1	0
	DMACR3	DMACR2	DMACR1	DMACR0			DMAEN	
Access	R/W	R/W	R/W	R/W			R/W	
Reset	0	0	0	0			0	

Bits 10:8 – DMABL[2:0] DMA System RAM Buffer Length

Bit 7 – DMACR3 DMA CORE3 Enable

Bit 6 – DMACR2 DMA CORE2 Enable

Bit 5 – DMACR1 DMA CORE1 Enable

Bit 4 – DMACR0 DMA CORE0 Enable

Bit 1 – DMAEN DMA Enable

38.7.20 APB FIFO Control Register

Name: PFFCTRL
Offset: 0x0E4
Reset: 0x00000000
Property: RW

The ADC FIFO is useful in applications that stream out ADC data at very high transfer rates to relieve CPU bandwidth. Individual high data rate ADC result interrupts and CPU reads may slow bus access transfer requests.

Table 38-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								PFFRDYDM
Access								R/W
Reset								0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PFFCR3	PFFCR2	PFFCR1	PFFCR0			PFFEN	
Access	R/W	R/W	R/W	R/W			R/W	
Reset	0	0	0	0			0	

Bit 16 – PFFRDYDM DMA APB FIFO Data Ready

Value	Name	Description
0	CTLINTFLAG_PFFHFUL	Selects CTLINTFLAG.PFFHFUL for the ADC DMA PFFRDY trigger signal to the DMAC
1	CTLINTFLAG_PFFRDY	Selects CTLINTFLAG.PFFRDY for the ADC DMA PFFRDY trigger signal to the DMAC

Bit 7 – PFFCR3 APB CORE 3 FIFO Enable

Bit 6 – PFFCR2 APB CORE 2 FIFO Enable

Bit 5 – PFFCR1 APB CORE 1 FIFO Enable

Bit 4 – PFFCR0 APB CORE 0 FIFO Enable

When PFFEN = 1, setting this bit enables the conversion output data of any channel k to be stored into the optional data FIFO.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 1 – PFFEN APB FIFO Enable

When the FIFO is disabled no data is being saved into the FIFO and the its logic is being kept in reset state.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Value	Description
0	FIFO is disabled
1	FIFO is enabled

38.7.21 CORE SYNC Busy Status Register

Name: SYNCBUSY
Offset: 0x0E8
Reset: 0x00000000
Property: R

Notes: The following GCLK-clocked registers are “Enable Write Protected”:

- CTRLC
- CTRLD
- CORCTRL
- CHNCFG1
- CHNCFG2
- CHNCFG3
- CHNCFG4
- CHNCFG5
- CALTRL
- FLTCTRL

Therefore, they do not require a SyncBusy bit. These registers are write-disabled when the CTRLA.ENABLE bit is set to enable the ADC. They can only be changed when the ADC is disabled (CTRLA.ENABLE = 0). The user must completely configure the ADC and then enable the ADC by setting CTRLA.ENABLE = 1.

Table 38-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
Access						CTRLB	ENABLE	SWRST
Reset						R	R	R
						0	0	0

Bit 2 – CTRLB CTRLB sync busy

For the GCLK-based register CTRLB:

When Hardware sets this bit, no writes are permitted to the CTRLB register.

Bit 1 – ENABLE Enable bit Sync Busy

For GCLK-based register bit CTRLA.ENABLE:

When Hardware sets this bit, no writes are permitted to the CTRLA.ENABLE register.

Bit 0 – SWRST Software Reset Sync Busy

Notes:

1. Typically, when the SWRST is written, the bit is auto-cleared the next APB clock cycle after. However, the SYNCBUSY.SWRST bit is set and stays set until the reset in the GCLK domain is completed. Therefore, the user must poll the SYNCBUSY register to know when the operation is complete.
2. Care must be taken during the APB reset phase, because potentially the external clock (GCLK) may not present.
3. During a SWRST, access to registers or bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

38.7.22 DMA Interrupt Enable Clear

Name: DMAINTENCLR
Offset: 0x0F0
Reset: 0x00000000
Property: RW

Table 38-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								SOVFL
Access								R/W
Reset								0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	RBF[3:0]				RAF[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – SOVFL Synchronizer Overflow

Bits 7:4 – RBF[3:0] Ram Buffer B Full

Bits 3:0 – RAF[3:0] Ram Buffer A Full

38.7.23 DMA Interrupt Enable Set

Name: DMAINTSET
Offset: 0x0F4
Reset: 0x00000000
Property: RW

Table 38-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
								SOVFL
Access								R/W
Reset								0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	RBF[3:0]				RAF[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 16 – SOVFL Synchronizer Overflow

Bits 7:4 – RBF[3:0] Ram Buffer B Full

Bits 3:0 – RAF[3:0] Ram Buffer A Full

38.7.24 DMA Interrupt Flag and Status

Name: DMAINTFLAG
Offset: 0x0F8
Reset: 0x00000000
Property: RW

Table 38-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
							DMAERR	SOVFL
Access							R/HS/K	R/K
Reset							0	0
Bit	15	14	13	12	11	10	9	8
	EBF[3:0]				EAF[3:0]			
Access	R/HS/K	R/HS/K	R/HS/K	R/HS/K	R/HS/K	R/HS/K	R/HS/K	R/HS/K
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RBF[3:0]				RAF[3:0]			
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0

Bit 17 – DMAERR DMA Bus Error

Bit 16 – SOVFL Synchronizer overflow

Bits 15:12 – EBF[3:0] Ram Buffer B Overflow Error

Bits 11:8 – EAF[3:0] Ram Buffer A Overflow Error

Bits 7:4 – RBF[3:0] Ram Buffer B Full

Bits 3:0 – RAF[3:0] Ram Buffer A Full

38.7.25 CORE Controller Interrupt Enable Set

Name: CTLINTENSET
Offset: 0x0FC
Reset: 0x00000000
Property: RW

Note: Writing a zero to these bits has no effect. Writing a one to these any of these bits will SET the ENABLE bit.

Note: A read of this register provides whether the Interrupt is Enabled (bit=1) or Disabled (bit=0), i.e. a write of a 1 to the bit then a read of the bit will return the interrupt is enabled (=1).

Table 38-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	VREFRDY	VREFUPD			CRRDY3	CRRDY2	CRRDY1	CRRDY0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 11 – PFFHFUL APB FIFO Half Full

Writing a 1 to this bit will enable the ADC FIFO Half Full interrupt request.
Reading this bit returns whether the PFFHFUL interrupt is enabled (1 = enabled).

Bit 10 – PFFRDY APB FIFO Ready

Writing a 1 to this bit will enable the ADC FIFO Data Ready interrupt request.
Reading this bit returns whether the PFFRDY interrupt is enabled (1 = enabled).

Bit 9 – PFFOVF APB FIFO Overflow

Writing a 1 to this bit will enable the ADC FIFO Overflow Error interrupt request,
Reading this bit returns whether the PFFOVF interrupt is enabled (1 = enabled).

Bit 8 – PFFUNF APB FIFO Underflow

Writing a 1 to this bit will enable the ADC FIFO Read Underflow Error interrupt request.
Reading this bit returns whether the PFFUNF interrupt is enabled (1 = enabled).

Bit 7 – VREFRDY VREF Ready

Writing a 1 to this bit will Enable the ADC Voltage Reference Ready as an interrupt request.
Reading this bit returns whether the VREFRDY interrupt is enabled (1 = enabled).

Bit 6 – VREFUPD VREF update

Writing a 1 to this bit will Enable the ADC Voltage Reference Ready Updated as an interrupt request.
Reading this bit returns whether the VREFUPD interrupt is enabled (1 = enabled).

Bit 3 – CRRDY3 Core 3 Ready

Bit 2 – CRRDY2 Core 2 Ready

Bit 1 – CRRDY1 Core 1 Ready

Bit 0 – CRRDY0 Core 0 Ready

Writing a 1 to this bit will enable the Core 0 Ready as an interrupt request.
Reading this bit returns whether the CRRDY0 interrupt is enabled (1 = enabled).

38.7.26 CORE Controller Interrupt Enable Clear

Name: CTLINTENCLR
Offset: 0x100
Reset: 0x00000000
Property: RW

Note: Writing a zero to these bits has no effect. Writing a one to these any of these bits will CLEAR the ENABLE bit.

Note: A read of this register provides whether the Interrupt is Enabled (bit=1) or Disabled (bit=0), i.e. a write of a 1 to the bit then a read of the bit will return the interrupt is disabled (=0).

Table 38-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	VREFRDY	VREFUPD			CRRDY3	CRRDY2	CRRDY1	CRRDY0
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit 11 – PFFHFUL APB FIFO Half Full

Writing a 1 to this bit will disable the ADC FIFO Half Full as an interrupt request.
Reading this bit returns whether the PFFHFUL interrupt is enabled (1 = enabled).

Bit 10 – PFFRDY APB FIFO Ready

Writing a 1 to this bit will disable the ADC FIFO Data Ready as an interrupt request.
Reading this bit returns whether the PFFRDY interrupt is enabled (1 = enabled).

Bit 9 – PFFOVF APB FIFO overflow

Writing a 1 to this bit will disable the ADC FIFO Overflow Error as an interrupt request,
Reading this bit returns whether the PFFOVF interrupt is enabled (1 = enabled).

Bit 8 – PFFUNF APB FIFO underflow

Writing a 1 to this bit will disable the ADC FIFO Read Underflow Error as an interrupt request.
Reading this bit returns whether the PFFUNF interrupt is enabled (1 = enabled).

Bit 7 – VREFRDY VREF Ready

Writing a 1 to this bit will disable the ADC Voltage Reference Ready as an interrupt request.
Reading this bit returns whether the VREFRDY interrupt is enabled (1 = enabled).

Bit 6 – VREFUPD VREF Update

Writing a 1 to this bit will disable the ADC Voltage Reference Ready Updated as an interrupt request.
Reading this bit returns whether the VREFUPD interrupt is enabled (1 = enabled).

Bit 3 – CRRDY3 Core Ready 3 Disable

Bit 2 – CRRDY2 Core Ready 2 Disable

Bit 1 – CRRDY1 Core Ready 1 Disable

Bit 0 – CRRDY0 Core Ready 0 Disable

Writing a 1 to this bit will disable the Core n Ready as an interrupt request.
Reading this bit returns whether the CRRDY interrupt is enabled (1 = enabled).

38.7.27 CORE Controller Interrupt Flags

Name: CTLINTFLAG
Offset: 0x104
Reset: 0x00000000
Property: RW

Table 38-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					PFFHFUL	PFFRDY	PFFOVF	PFFUNF
Access					R	R	R/K	R/K
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	VREFRDY	VREFUPD			CRRDY[3:0]			
Access	R	R/K			R	R	R	R
Reset	0	0			0	0	0	0

Bit 11 – PFFHFUL APB FIFO Half Full

This bit is set when the FIFO is at least half full of data to be read. When set, this bit enables this condition to trigger the ADC interrupt if the corresponding bit in CTLINTSET is set. This bit is cleared by hardware when the FIFO output data has been read and there is less than a half full FIFO left to be read.

This bit is NOT reset by software writing a 1 to it, it is only reset by hardware.

Bit 10 – PFFRDY APB FIFO Ready

This bit is set when the FIFO has data to be read. When set, this bit enables the trigger of the ADC interrupt if the corresponding bit in CTLINTSET is set.

This bit is cleared by hardware when the FIFO output data in has been read and there is no additional data ready in the FIFO (that is the APB FIFO is empty). This bit is NOT reset by software writing a 1 to it.

Bit 9 – PFFOVF APB FIFO overflow

This bit is set by hardware when the FIFO is full and new output data overwrites data to be read. When set, this bit enables this condition to trigger an ADC interrupt if the corresponding bit in CTLINTSET is set.

This bit is reset by software writing a 1 to it.

Bit 8 – PFFUNF APB FIFO underflow

This bit is set by hardware when the FIFO is empty. When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.

This bit is reset by software writing a 1 to it.

Bit 7 – VREFRDY VREF Ready

Hardware sets and clears this bit according to the status of the ADC Voltage Reference. ADC Conversion Data is valid only after VREFRDY is set by hardware, so the ADC interrupt service routine in charge of data processing should always check first that VREFRDY is set to ensure the data validity. This bit will only be updated to the VREFRDY condition set or cleared when the CTRLA.ENABLE is on. Therefore, once the ISR has verified that VREFRDY=1, it should disable the corresponding interrupt by setting CTLINTENCLR.VREFRDY = 1 to prevent continuous firing of the ISR. Alternately, the software setup routine for the ADC can wait in a while(1) loop, polling VREFRDY until it goes high.

Bit 6 – VREFUPD VREF update

This bit is set by hardware on both the positive and negative edges of the bit VREFRDY. This means that the hardware will set this bit when the ADC Voltage Reference is ready, but also when it fails, that is on any change. When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.

This bit is reset by software writing a 1 to it. It is NOT cleared by a software read.

Software must read the value of VREFRDY to ascertain if the ADC analog reference circuits are in order or not when the CPU is servicing the interrupt prompted by VREFUPD. The ADC Voltage Reference analog signals are required to be ready during operation of the ADC. If an ADC Voltage Reference fault is detected, the ADC module must be re-calibrated. Most likely an ADC Reference Voltage fault is caused by a brown-out of the analog Vdd supply.

Hardware sets this bit to zero when CTRLA.ENABLE=0.

Bits 3:0 – CRRDY[3:0] Core Ready

38.7.28 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x120
Reset: 0x00000000
Property: RW

Note: Writing a zero to any of these bits has no effect, but writing a one to these bits will CLEAR the ENABLE bit.

A read of this register provides the current status of interrupts, that is whether each interrupt is enabled (bit = 1) or disabled (bit = 0).

Table 38-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHRDY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHRDY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					EOSRDY	CHNERRC	FLTRDY	CHRDYC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SOVFL			CMPHIT				
Access	R/W			R/W				
Reset	0			0				

Bits 31:16 – CHRDY[15:0] Channel Ready Disable

Writing a 1 to bit will k disable the data ready flag for channel k as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 11 – EOSRDY Endo of Scan Disable

Writing a 1 to this bit will disable the flag bit EOSRDY as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 10 – CHNERRC Channel Overwrite Error Disable

Writing a 1 to this bit will disable the flag bit CHNERRC as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

Bit 9 – FLTRDY Filter Ready Disable

Writing a 1 to this bit will disable the filter ready flag for Filter as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 8 – CHRDYC Core Current Channel Disable

Writing a 1 to this bit will disable the flag bit CHRDYC as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 7 – SOVFL Synchronizer Overflow Disable

Writing a 1 to this bit will disable the flag SOVFL as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 4 – CMPHIT Compare Hit Disable

Writing a 1 to this bit will disable the comparator hit flag for Comparator n as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

38.7.29 Interrupt Enable Set

Name: INTENSET
Offset: 0x124
Reset: 0x00000000
Property: RW

Note: Writing a zero to any of these bits has no effect, but writing a one to these bits will SET the ENABLE bit.

A read of this register provides the current status of interrupts, i.e., whether each interrupt is enabled (bit=1) or disabled (bit=0).

Table 38-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHRDY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHRDY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					EOSRDY	CHNERRC	FLTRDY	CHRDYC
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SOVFL			CMPHIT				
Access	R/W			R/W				
Reset	0			0				

Bits 31:16 – CHRDY[15:0] Channel Ready Enable

Writing a 1 to bit k will enable the data ready flag for Core n channel k as an interrupt request. CHRDY[k] is defined only for k = 0,1,2,...,(Sn-1).

Note:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 11 – EOSRDY End of Scan Enable

Writing a 1 to this bit will disable the flag bit EOSRDY as an interrupt request.

Note:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 10 – CHNERRC Channel Overwrite Enable

Writing a 1 to this bit will enable the flag bit CHNERRC as an interrupt request.

Note:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 9 – FLTRDY Filter Ready Enable

Writing a 1 to this bit will enable the filter ready flag for Filter as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 8 – CHRDYC Current Channel Ready Enable

Writing a 1 to this bit will enable the flag bit CHRDYC as an interrupt request.

Note:

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 7 – SOVFL Synchronizer Overflow Enable

Writing a 1 to this bit will enable the flag SOVFL as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

Bit 4 – CMPHIT Compare Hit Enable

Writing a 1 to this bit will enable the comparator hit flag for Comparator n as an interrupt request.

Note:

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

38.7.30 Interrupt Flags

Name: INTFLAG
Offset: 0x128
Reset: 0x00000000
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 38-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CHRDY[15:8]							
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CHRDY[7:0]							
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CRDYID[3:0]				EOSRDY	CHNERRC	FLTRDY	CHRDYC
Access	R/K	R/K	R/K	R/K	R/K	R/K	R/K	R/K
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	SOVFL			CMPHIT	CMPINTID[3:0]			
Access	R/K			R/K	R/K	R/K	R/K	R/K
Reset	0			0	0	0	0	0

Bits 31:16 – CHRDY[15:0] Channel Ready

CHRDY[k] = 1 indicates that ADC has completed its last A/D conversion for channel k.

Note: If INTENSET.CHRDY[k] is set, then CHRDY[k] = 1 will trigger an ADC interrupt. CHRDY[k] is reset by software writing a 1 to it.

Bits 15:12 – CRDYID[3:0] Channel Ready ID

The value of CRDYID indicates the input channel index, k , that ADC has just completed in its current scan. This is for information only since these bits cannot be used to trigger an ADC interrupt. (CHRDY[k] are intended for that purpose.)

These bits are reset by software writing a 1 to them.

Bit 11 – EOSRDY End of Scan Ready

This bit is set by hardware at the end of the scan of all channels included in the scan performed by ADC in response to a SINGLE event of the Scan Trigger 0 (STRIG0).

If INTENSET.EOSRDY is set, then EOSRDY = 1 will trigger an ADC interrupt.

This bit is reset by software writing a 1 to it.

Bit 10 – CHNERRC Channel Overwrite Error

When set, this bit indicates that the ADC has completed its last A/D conversion for channel CRDYID[3:0], but at the time CRDYID[3:0] was updated, the status bit CHRDYC was still set, which indicates that the software may not have had the time to read the previous data, which may be now lost.

If INTENSET.CHNERRC is set, then CHNERRC = 1 will trigger an ADC interrupt.

This bit is reset by software writing a 1 to it.

Bit 9 – FLTRDY Filter Ready

When set, this bit indicates that the digital filter has issued a new output sample for the input channel defined by FLTCTRL.FLTCHNID.

If INTENSET.FLTRDY is set, then FLTRDY = 1 will trigger an ADC interrupt.

This bit is reset by software writing a 1 to it.

Bit 8 – CHRDYC Current Channel Ready

0 = ADC busy or idle.

1 = When set, this bit signifies that the ADC has completed its current A/D conversion for the channel identified in CRDYID[3:0].

Notes:

1. If INTENSET.CHRDYC is set, then CHRDYC = 1 will trigger an ADC interrupt.
2. This bit is reset by software writing a 1 to it.

Bit 7 – SOVFL Synchronizer Overflow

When set this bit signifies the ADC data was lost due to a slow APB_CLK.

If INTENSET.SOVFL is set, then SOVFL = 1 will trigger an ADC interrupt.

This bit is reset by software writing a 1 to it.

Bit 4 – CMPHIT Compare Hit

When set, this bit signifies that the Digital Comparator associated with ADC has issued a condition hit interrupt for channel identified in CMPINTID[5:0].

If INTENSET.CMPHIT is set, then CMPHIT = 1 will trigger an ADC interrupt.

This bit is reset by software writing a 1 to it.

Bits 3:0 – CMPINTID[3:0] Compare Channel ID

When set, this signifies that the ADC Channel ID for which the digital comparator has issued the condition hit interrupt (CMPHIT=1). These bits cannot be used as an interrupt request flag. This is for information only. (CMPHIT is intended for that purpose.)

This bit is reset by software writing a 1 to it.

38.7.31 Debug Control Register

Name: DBGCTRL
Offset: 0x168
Reset: 0x00000000
Property: RW

Table 38-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								R/W
Reset								0

Bit 0 – DBGRUN Debug Running State

This bit is not affected by software reset and should not be changed by software while the ADC is enabled.

Note: During debug operation with DBGRUN=0 the ADC captures input trigger events and on exit from the debug the ADC will (based on priority) perform conversions on the captured event or events.

Value	Description
0	The ADC is halted when the CPU is halted in debug mode.
1	The ADC continues normal operation when the CPU is halted in debug mode.

39. Analog Comparator (AC)

39.1 Overview

The Analog Comparator (AC) consists of two individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change. The two comparators can combine to support a window function, discussed below.

Hysteresis is user selectable to achieve the optimal operation for each application.

By default, the input selection includes four shared analog port pins and several internal signals. The flexible internal architecture allows additional input selections. For details, refer to each chip-specific documentation. Each comparator output state can also be output on a pin for use by external devices.

The comparators are grouped in pairs on each port. The AC peripheral implements one, called Comparator 0 (COMP0) and Comparator 1 (COMP1). The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.



Important: If $AVDD < 2.5V$, users must enable analog charge pumps in `SUPC.VREGCTRL.CPEN[1:0]`.

39.2 Features

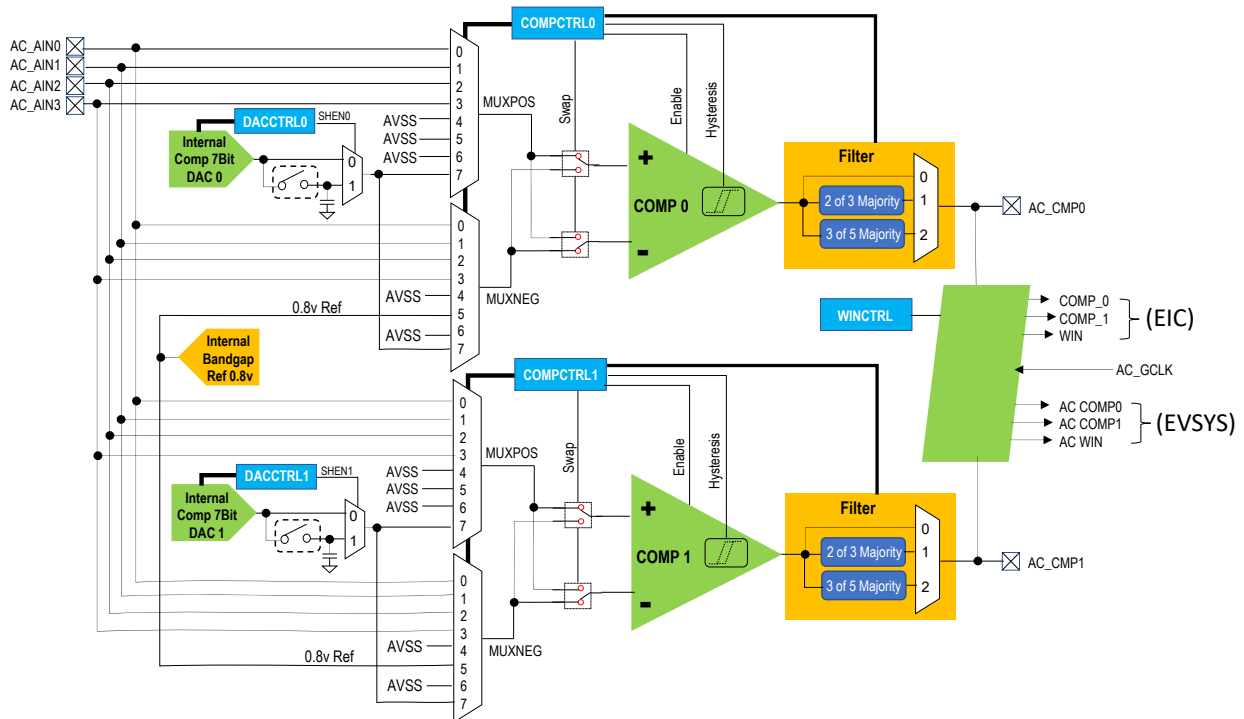
The following are key features of the AC module:

- Four individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis
 - 4-levels , Always ON
- Analog comparator outputs available on pins
 - Asynchronous or synchronous
- Default input selection:
 - 4 pins selectable for positive or negative inputs
 - Ground (for zero crossing)
 - Supports 7-bit dedicated DACs for each comparator, enabling up to 2^{12} -level programmable VDD scaler
 - Extensions available to connect additional internal or external pins on each device
- Interrupt generation on:
 - Rising or falling edge
 - Toggle
 - End of comparison
- Window function interrupt generation on:
 - Signal above window
 - Signal inside window
 - Signal below window
 - Signal outside window

- Event generation on:
 - Comparator output
 - Window function inside/outside window
- Optional digital filter on comparator output
- Low-power options:
 - Single-shot support
 - Continuous or sampling DAC operating mode
- Ability to delay the comparator output by n-cycle of controller clock

39.3 Block Diagram

Figure 39-1. Analog Comparator Block Diagram



39.4 Analog Connections

Each comparator has minimum four I/O pins that can be used as analog inputs. Each pair of comparators shares four pins. These pins must be configured for analog operation before using them as comparator inputs.

The Analog comparator embeds additional capabilities, enabling the option to connect more I/O pins from device to device, or other internal references. This selection is device specific, and for further details refer to the device specification.

Any internal reference source, such as a bandgap voltage reference or internal DAC must be configured and enabled prior to its use as a comparator input. Any external reference must be enabled and ready to be used, before the comparator is enabled.

39.5 Functional Description

39.5.1 Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as a bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive, and '0' otherwise.

The individual comparators can be used independently (Normal mode) or paired to form a window comparison (Window mode).

39.5.2 Basic Operation

39.5.2.1 Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See *Starting a Comparison* for more details.
- Select the desired hysteresis with COMPCTRLx.HYST. See *Input Hysteresis* for more details.
- Write COMPCTRLx.SPEED.
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See *Selecting Comparator Inputs* for more details.
- Select the filtering option with COMPCTRLx.FLEN
- Select standby operation with Run in Standby bit (CTRLA.RUNSTDBY) and (COMPCTRLx.AC_RUNSTDBY)

The individual comparators are enabled by writing a '1' to the Enable bit in the COMPENABLE.ENABLEx. The individual comparators are disabled by writing a '0' to COMPENABLE.ENABLEx. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.

39.5.2.2 Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay (tSTARTUP) is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in the Electrical Characteristics. During the start-up time, the COMP output is not available. For details on how the start-up time is calculated, refer to COMPCTRLx.SUT description.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge), or at the end of the comparison. An end-of-comparison interrupt can be used with the single-shot mode to chain further events in the system, regardless of the state of the comparator outputs. The Interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

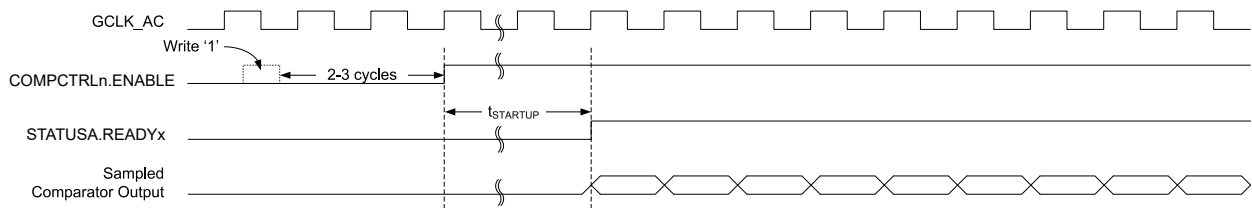
39.5.2.2.1 Continuous Measurement

Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status A register (STATUSA.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPENABLE.ENABLEx bit is written to zero. The start-up time applies only to the first comparison.

In continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the following figure.

Figure 39-2. Continuous Measurement Example



For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is then disabled again automatically, unless configured to wake up the system from sleep.

39.5.2.2.2 Single-Shot

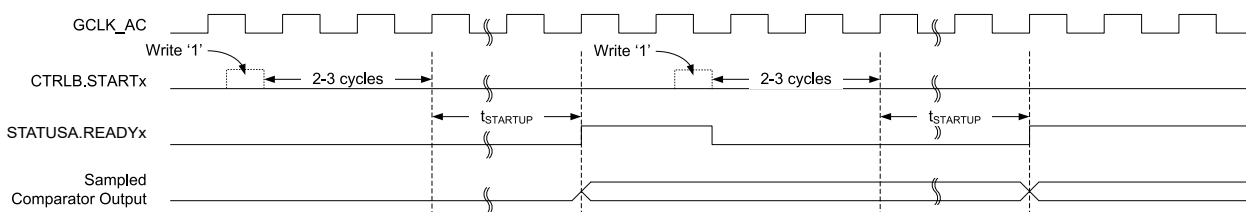
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status A register (STATUSA.READYx). STATUSA.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in the following figure.

Figure 39-3. Single-Shot Example



For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the startup time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake up the system from sleep.

39.5.3 Selecting Comparator Inputs

Each comparator has one positive and one negative input. These inputs can come from one of the external input pins (AINx) or from one of the several internal reference voltage sources. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

39.5.4 Window Operation

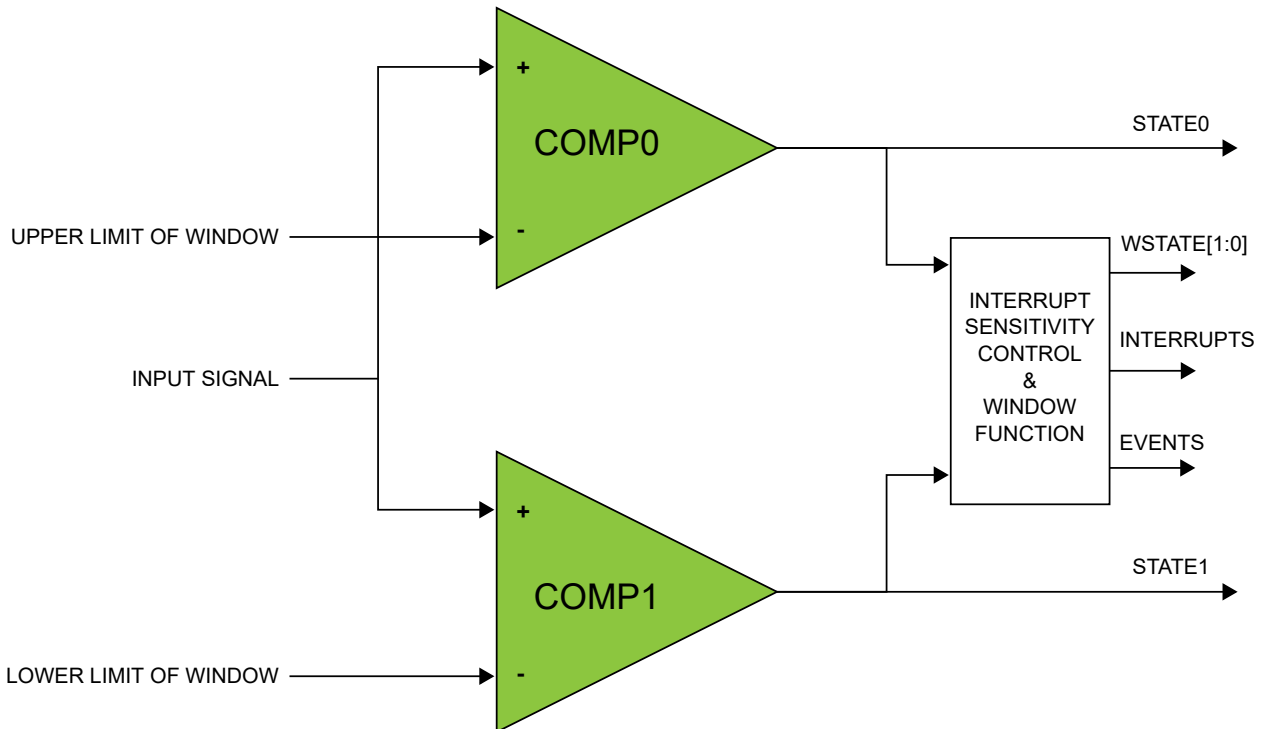
Each comparator pair can be configured to work together in Window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

To physically configure the pair of comparators for Window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In the following figure, COMP0 defines the upper limit and COMP1 defines the lower limit of the window, as shown but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during Window mode.

When the comparators are configured for Window mode and Single-shot mode, measurements are performed simultaneously on both comparators when both comparators are ready. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise either peripheral event can start a measurement.

Figure 39-4. Comparators in Window Mode



39.5.5 Internal DAC Operation

Each Analog Comparator pair, includes two DACs, each connected to its respective ACx.

The DACx is enabled when the analog comparator is enabled (`COMPENABLE.ENABLEx = 1`), and the DAC is used as positive or negative input for the respective comparator (`MUXNEG = INTDAC` or `MUXPOS = INTDAC`).

The DAC is disabled when the analog comparator is disabled (`COMPENABLE.ENABLEx = 0`).

The DAC configuration registers are cleared when the analog comparator is reset (`CTRLA.SWRST = 1`).

The DAC generates a reference voltage that is a fraction of the device's supply voltage, with 2^7 levels. One independent DAC channel is dedicated for each comparator. The voltage of each channel is selected by the Value x bit field in the DACCTRL registers (`DACCTRL.VALUEx`).

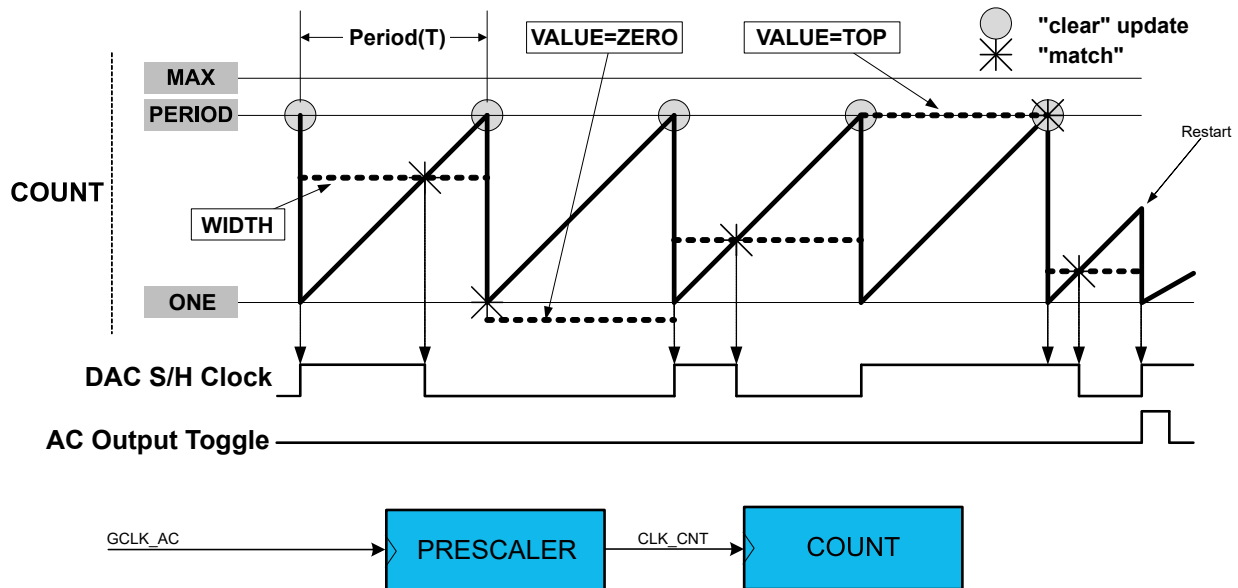
The DAC can be enabled in continuous operating mode (`DACCTRL2.SHENx = 0`), or in sample mode (`DACCTRL2.SHENx = 1`). When set in continuous mode, no clocks are required for operation. In sampling mode, each DAC controller includes a dedicated counter, generating the sampling clock. The counter operation is started when the analog comparator is ready (`STATUSA.READYx = 1`), and stopped when the analog comparator is disabled. As a consequence, in single shot mode, the DAC is enabled in continuous mode when the AC is enabled (`COMPENABLE.ENABLEx = 1`), and switches to the sample and hold operation when the comparator is ready.

Note: the DAC S/H operating mode and AC single-shot mode are two independent operations, and not linked together. Both S/H and continuous DAC operating modes can be selected when the AC is used in either continuous or single-shot mode. The operating modes must be selected by the application, depending on power consumption and/or response time requirements.

The counter counts up and restarts when the software PERIOD programmable value (`CTRLC.PER`) is reached. The counter operation is stopped and the internal counter is cleared when the sampling and hold operation is not required. When the AC output toggles, the counter is restarted.

The sampling clock is set when the counter operation starts, and cleared when the counter reaches the WIDTH software programmable value (CTRLC.WIDTH).

Figure 39-5. Sample and Hold Clock Generation



39.5.6 Input Hysteresis

Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other. Hysteresis is always enabled for each comparator. The level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST).

39.5.7 Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bit selects the amount of bias current provided to the comparator, and as such will also affect the start-up time.

39.5.8 Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note: that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For Continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Continuous Mode Filtering. For Single-shot mode, the comparison completes after the Nth filter sample, as shown in Single-Shot Filtering.

Figure 39-6. Continuous Mode Filtering

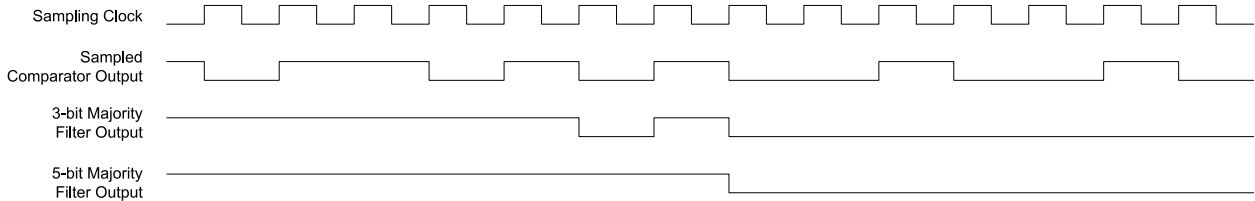
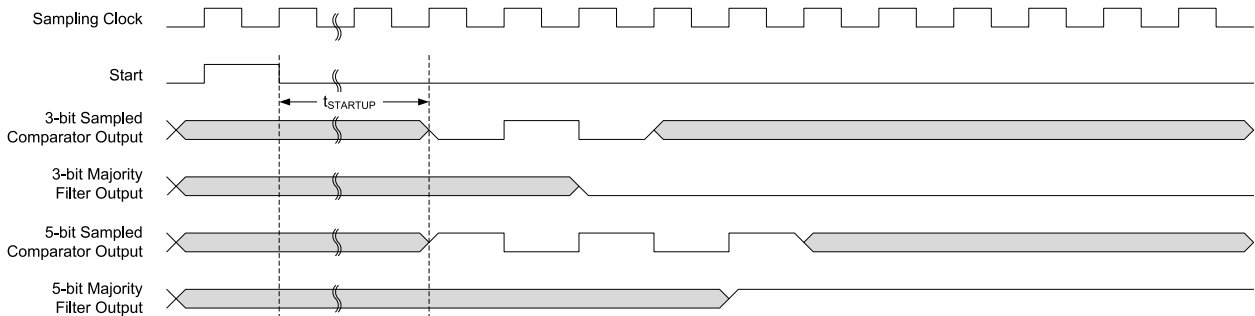


Figure 39-7. Single-Shot Filtering



During Sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during Sleep modes, or the resulting interrupt/event may be generated incorrectly.

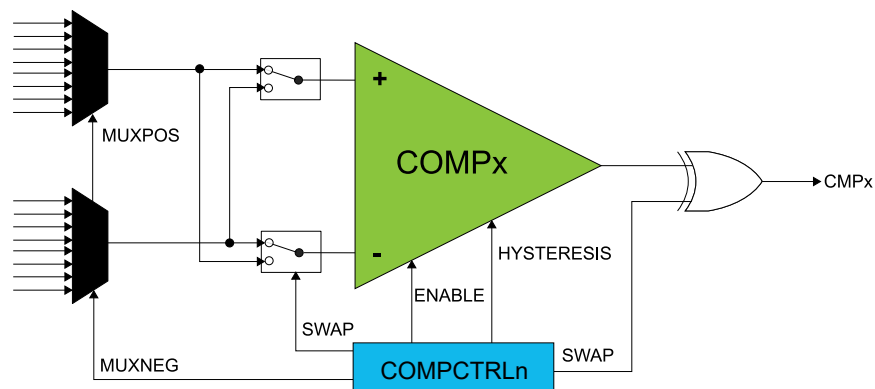
39.5.9 Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control n register (COMPCTRLn.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

39.5.10 Inputs Swapping

The Swap bit in the Comparator Control registers (COMPCTRLn.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in the figure below. As part of the input selection, COMPCTRLn.SWAP can be changed only while the comparator is disabled.

Figure 39-8. Input Swapping



39.5.11 Inputs Swapping

The analog comparator output can be blanked using a blanking source input from the event system, including choosing the blanking window (delay and size) in addition to enabling blanking. The Comparator blanking window will always be delayed up to a maximum of 3 clocks (sync flops) from the rising of the event input to the comparator. The Blanking Window then can be Delayed by the amount of cycles in the Register 1-17: CMPBWINDLY0k - Comparator 0 output blanking WINDOW DELAY Register and the blanking window size is defined by Register 1-16: CMPBWIN0k - Comparator 0 output blanking WINDOW Register. These two registers allow for the Blanking Window to be adjusted by the user to their specific needs by n cycles of the defined GCLK.

If another blanking event comes before blanking window expires, then that event will be ignored. The blanking source is basically ANDed (see Figure 2-9: "Comparator Output Signal Blanking Example") with the comparator output, resulting in a final comparator output which then can be filtered or not based on the filter enable.

Figure 39-9. Output Blanking

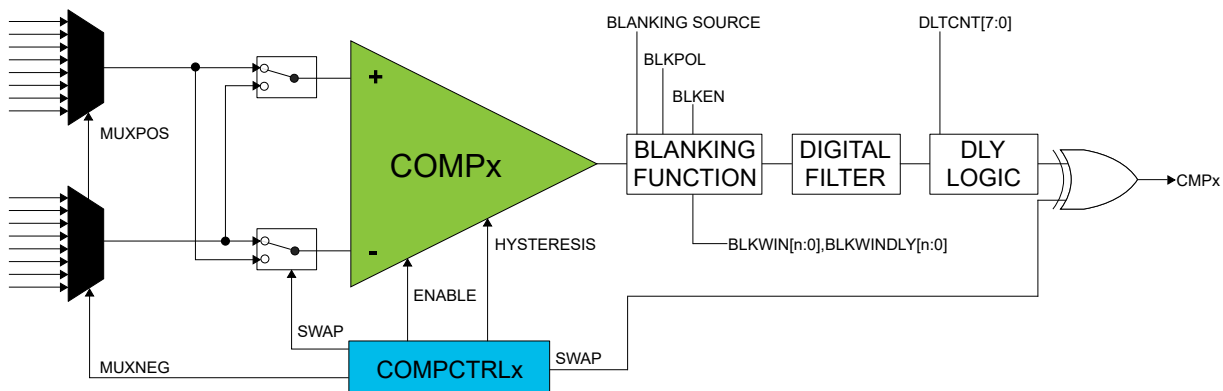
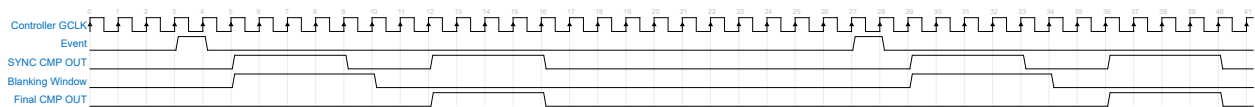


Figure 39-10. Comparator Output Signal Blanking Example

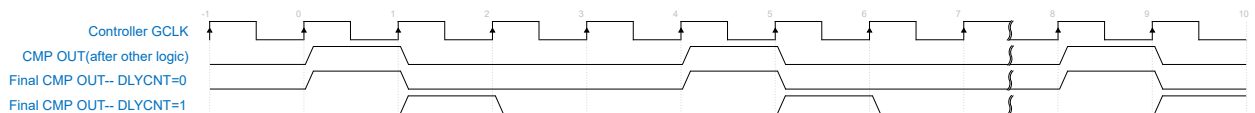


39.5.12 Comparator Output Delay

The analog comparator output can be delayed by n-cycles of the control clock by using the calibration counter n-cycle values in the CMPOUCFGn register. The user has the ability to disable or re-enable this delay if desired in the COMPCTRL0n and COMPCTRL1k registers, but cannot write over the calibration values provided in the CMPOUCFGn register.

If single shot mode and output delay logic is enabled, the single comparison output will be delayed by n clock cycles (CMPOUCFGn.DLTCNT).

Figure 39-11. Comparator Output Delay Example



39.5.13 Interrupts

The AC has the following interrupt sources:

- Comparator (COMPx, x = {0,1}): Indicates a change in comparator status.

- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLn.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSELx).

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the AC is reset. See INTFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

39.5.14 Events

The AC can generate the following output events:

- Comparator (COMPx, x={0,1}): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (STARTx, x={0,1}): Start a comparison

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring the event system.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

39.5.15 Sleep Mode Operation

The Run in Standby bits in the Comparator n Control registers (COMPCTRLn.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in the following table.

Table 39-1. Sleep Mode Operation

COMPCTRLn.MODE	RUNSTDBY=0	RUNSTDBY=1
0 (Continuous)	GCLK_AC stopped, COMPx disabled	GCLK_AC stopped or running (depends on filtering options), COMPx enabled
1 (Single-shot)	GCLK_AC stopped, COMPx disabled	GCLK_AC stopped, COMPx enabled only when triggered by an input event

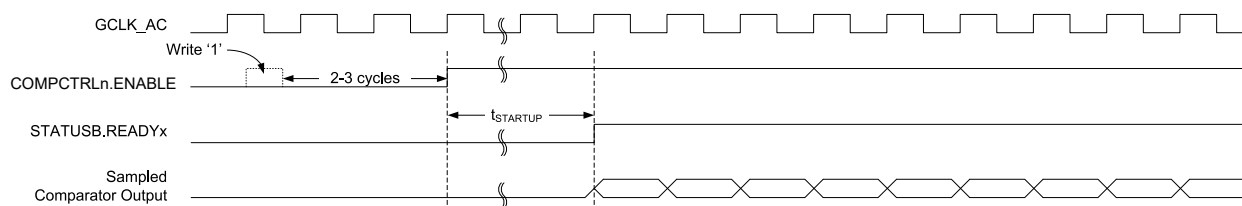
The Run in Standby bits in the Comparator n Control registers (COMPCTRLn.RUNSTDBY) also controls the behavior of the internal DAC during standby sleep mode, if the DAC is selected a positive or negative multiplexor input (COMPCTRLn.MUXPOS = INTDAC, COMPCTRLn.MUXNEG = INTDAC). When the bit is zero, the DAC is disabled during sleep, but maintains its current configuration. When the bit is one, the DAC continues to operate during sleep. Note that when RUNSTDBY is zero, the DAC blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

39.5.15.1 Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and the filtering options are disabled, the GCLK_AC is disabled during sleep and the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device; otherwise GCLK_AC is disabled until the next edge detection.

If the filtering is enabled, the GCLK_AC is kept running, the comparator will remain continuously enabled and will function synchronously. The current state of the comparator is synchronously monitored for changes. If an edge matching the interrupt condition is found and the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device.

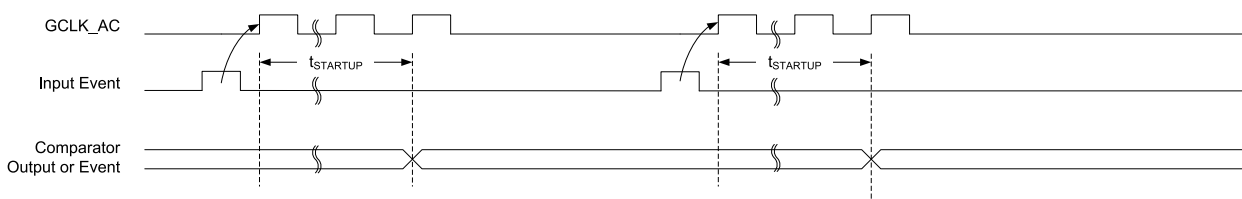
Figure 39-12. Continuous Mode SleepWalking



39.5.15.2 Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in the following figure. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

Figure 39-13. Single-Shot SleepWalking



39.6 Peripheral Dependencies

Table 39-2. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
AC	0x4482A000	132 : COMP0/COMP1/WIN0	APB: CLKMSK[1] MASK18	PCHCTRL[34]	40 STATUS1 PERID8		<u>Generators:</u> COMP0: CHANNEL[78] COMP1: CHANNEL[79] WIN0: CHANNEL[80] <u>Users:</u> SOC0: USER[49],Resynch SOC1: USER[50],Resynch	PD_CORE_SW

39.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x04	CTRLB	31:24								
		23:16								
		15:8								
		7:0								
0x08	CTRLC	31:24								
		23:16								
		15:8								
		7:0								
0x0C	EVCTRL	31:24								
		23:16								
		15:8								
		7:0								
0x10	INTENCLR	31:24								
		23:16								
		15:8								
		7:0								
0x14	INTENSET	31:24								
		23:16								
		15:8								
		7:0								
0x18	INTFLAG	31:24								
		23:16								
		15:8								
		7:0								
0x1C	STATUSA	31:24								
		23:16								
		15:8								
		7:0								
0x20	STATUSB	31:24								
		23:16								
		15:8								
		7:0								
0x24	DBGCTRL	31:24								
		23:16								
		15:8								
		7:0								
0x28	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0								
0x2C	Reserved	31:24								
...		23:16								
0x2F		15:8								
		7:0								
0x30	COMPCTRL0	31:24								
		23:16								
		15:8								
		7:0								
0x34	COMPCTRL1	31:24								
		23:16								
		15:8								
		7:0								
0x38	DACCTRL	31:24								
		23:16								
		15:8								
		7:0								

Register Summary (continued)										
Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x3C	WINCTRL	31:24								
		23:16								
		15:8								
		7:0						WINTSEL[1:0]		WEN

39.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 39-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							RW	W
Reset							0	0

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

Value	Description
0	The AC is disabled.
1	The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE).

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the AC to their initial state, and the AC will be disabled.

Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.

Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete.

CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

Note: To avoid spurious interrupts from enable/disable cycles, use the SWRST bit to reset the comparator module.

Value	Description
0	There is no reset operation ongoing.
1	The reset operation is ongoing.

39.7.2 Control B

Name: CTRLB
Offset: 0x04
Reset: 0x00000000
Property: RW

Table 39-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	START[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – START[7:0] Comparator x Start Comparison

39.7.3 Control C

Name: CTRLC
Offset: 0x08
Reset: 0x00000000
Property: RW

Table 39-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CONFIG[3:0]				AIPMPEN	PRESCALER[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
			PER[9:4]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PER[3:0]						WIDTH[9:8]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0
Bit	7	6	5	4	3	2	1	0
	WIDTH[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – CONFIG[3:0] Configuration Extension
These bits are reserved.

Bit 27 – AIPMPEN Analog Input Charge Pump Enable

Bits 26:24 – PRESCALER[2:0] Prescaling Factor
These bits define the prescaling factor for the AC clock source (GCLK_AC) to generate the DAC sampling clock as shown in the following table.

Value	Name	Description
0x0	DIV1	Sampling rate is GCLK_AC (No division)
0x1	DIV2	Sampling rate is GCLK_AC/2
0x2	DIV4	Sampling rate is GCLK_AC/4
0x3	DIV8	Sampling rate is GCLK_AC/8
0x4	DIV16	Sampling rate is GCLK_AC/16
0x5	DIV32	Sampling rate is GCLK_AC/32
0x6	DIV64	Sampling rate is GCLK_AC/64
0x7	DIV128	Sampling rate is GCLK_AC/128

Value	Name	Description
0x0	DIV1	Sampling rate is GCLK_AC (No division)

Value	Name	Description
0x1	DIV2	Sampling rate is GCLK_AC/2
0x2	DIV4	Sampling rate is GCLK_AC/4
0x3	DIV8	Sampling rate is GCLK_AC/8
0x4	DIV16	Sampling rate is GCLK_AC/16
0x5	DIV32	Sampling rate is GCLK_AC/32
0x6	DIV64	Sampling rate is GCLK_AC/64
0x7	DIV128	Sampling rate is GCLK_AC/128

Bits 21:12 – PER[9:0] Sample and Hold Clock Period

These bits configure the sample and hold clock period. If PER is set to zero, no sample and hold DAC clock is generated.

Note: These bits are ignored if DACCTRLn.SHENn=0 (i.e. DAC continuous operation mode is enabled).

Bits 9:0 – WIDTH[9:0] Sample and Hold Clock Pulse Width

These bits configure the sample and hold clock pulse width. If WIDTH is set to zero, no sample and hold DAC clock is generated.

Note: These bits are ignored if DACCTRLn.SHENn=0 (i.e. DAC continuous operation mode is enabled).

39.7.4 Event Control

Name: EVCTRL
Offset: 0x0C
Reset: 0x00000000
Property: RW

Table 39-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	INVEI[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COMPEI[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					WINEO[3:0]			
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMPEO[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – INVEI[7:0] Inverted Event Input Enable x

Bits 23:16 – COMPEI[7:0] Comparator x Event Input Enable

Bits 11:8 – WINEO[3:0] Window x Event Output Enable

Bits 7:0 – COMPEO[7:0] Comparator x Event Output Enable

39.7.5 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x10
Reset: 0x00000000
Property: RW

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

Table 39-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						WIN[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – WIN[3:0] Window x Interrupt Enable

Bits 7:0 – COMP[7:0] Comparator x Interrupt Enable

39.7.6 Interrupt Enable Set

Name: INTENSET
Offset: 0x14
Reset: 0x00000000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

Table 39-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						WIN[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – WIN[3:0] Window x Interrupt Enable

Bits 7:0 – COMP[7:0] Comparator x Interrupt Enable

39.7.7 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x18
Reset: 0x00000000
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 39-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
						WIN[3:0]		
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 11:8 – WIN[3:0] Window x

Bits 7:0 – COMP[7:0] Comparator x

39.7.8 Status A

Name: STATUSA
Offset: 0x1C
Reset: 0x00000000
Property: R

Table 39-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WSTATE1[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WSTATE0[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	STATE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 16:23, 24:31, 32:39, 40:47 – WSTATE Window x Current State

Value	Name	Description
0	ABOVE	Signal is above window
1	INSIDE	Signal is inside window
2	BELOW	Signal is below window

Bits 7:0 – STATE[7:0] Comparator x Current State

39.7.9 Status B

Name: STATUSB
Offset: 0x20
Reset: 0x00000000
Property: R

Table 39-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	READY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – READY[7:0] Comparator x Ready

39.7.10 Debug Control

Name: DBGCTRL
Offset: 0x24
Reset: 0x00000000
Property: RW

Table 39-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								DBGRUN
Access								RW
Reset								0

Bit 0 – DBGRUN Debug Run

This bit is not reset by a software reset.

This bits controls the functionality when the CPU is halted by an external debugger.

Value	Description
0	The AC is halted when the CPU is halted by an external debugger. Any on-going comparison will complete.
1	The AC continues normal operation when the CPU is halted by an external debugger.

39.7.11 Synchronization Busy

Name: SYNCBUSY
Offset: 0x28
Reset: 0x00000000
Property: R

Table 39-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			WINCTRL[3:0]				COMPCTRL[7:6]	
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COMPCTRL[5:0]						ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 13:10 – WINCTRL[3:0] WINCTRL x Synchronization Busy

Bits 9:2 – COMPCTRL[7:0] COMPCTRL x Synchronization Busy

Bit 1 – ENABLE Enable Synchronization Busy

This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

Bit 0 – SWRST Software Reset Synchronization Busy

This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.

This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.

39.7.12 Pair n Comparator Control 0

Name: COMPCTRL
Offset: 0x30 + n*0x04 [n=0..1]
Reset: 0x00000000
Property: RW

Table 39-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SUT[5:0]					OUT[1:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FLEN[2:0]			HYST[1:0]			SPEED	SWAP
Access	R/W	R/W	R/W	R/W	R/W		R/W	R/W
Reset	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8
		MUXPOS[2:0]				MUXNEG[2:0]		
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset		0	0	0		0	0	0
Bit	7	6	5	4	3	2	1	0
		RUNSTDBY	INTSEL[1:0]		SINGLE		ENABLE	
Access		R/W	R/W	R/W	R/W		R/W	
Reset		0	0	0	0		0	

Bits 31:26 – SUT[5:0] Start-up Time

Each time a comparator is enabled, the comparison will be enabled after the startup time specified by these bits using the formula:

$$SUT = (20\mu s / ((2^{\text{PRESCALER}}) \times t(\text{GCLK\_AC})))$$

Refer to electrical specifications for minimum analog comparator start-up time required to initialize COMPCTRLn.SUT bits to.

Notes:

1. If COMPCTRLn.SINGLE=1 even if comparator COMPCTRLn.ENABLE=1 the comparator is turned off to conserve power. When CTRLB.STARTn is set by the user to start the comparison the comparator is enabled, after COMPCTRLn.SUT time has expired the comparison is made and then the comparator is again shut off to conserve power ready for the next time.
2. These bits can be written only while COMPCTRLn.ENABLE is zero.
3. Zero value is not allowed and can lead to unpredictable behavior.

Bits 25:24 – OUT[1:0] Output

These bits configure the output selection for comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is zero.

Note: If the asynchronous path is selected, the filter settings are ignored.

Note: If OUT = 0x2 = Sync, Filtering must be enabled, FLEN>0.

Value	Name	Description
0	OFF	The output of COMPn is not routed to the COMPn I/O port
1	ASYN	The asynchronous output of COMPn is routed to the COMPn I/O port
2	SYNC	The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port

Bits 23:21 – FLEN[2:0] Filter Length

These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

Note: If COMPCTRLn.OUT=0x1 (Asynchronous Mode), these bits are ignored.

Value	Name	Description
0	OFF	No filtering
1	MAJ3	3-bit majority function (2 of 3)
2	MAJ5	5-bit majority function (3 of 5)

Bits 20:19 – HYST[1:0] Hysteresis Level

These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN=1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE=0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.

Value	Name	Description
0	HYST10	10mV
1	HYST20	20mV
2	HYST40	40mV
3	HYST60	60mV

Bit 17 – SPEED Speed Selection

This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.

Note: High speed equates to higher operating current as well as faster response time.

Value	Name	Description
0	HIGH	High speed, high power
1	LOW	Low speed, low power

Bit 16 – SWAP Swap Inputs and Invert

This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is zero.

Value	Description
0	The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input.
1	The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input.

Bits 14:12 – MUXPOS[2:0] Positive Input Mux Selection

These bits select which input will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.

Value	Name	Description
0	PIN0	I/O pin 0
1	PIN1	I/O pin 1
2	PIN2	I/O pin 2
3	PIN3	I/O pin 3
4	INT0	Internal connection 0, device specific
5	INT1	Internal connection 1, device specific

Value	Name	Description
6	INT2	Internal connection 2, device specific
7	INTDAC	Internal DAC

Bits 10:8 – MUXNEG[2:0] Negative Input Mux Selection

These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.

Value	Name	Description
0	PIN0	I/O pin 0
1	PIN1	I/O pin 1
2	PIN2	I/O pin 2
3	PIN3	I/O pin 3
4	INT0	Internal connection 0, device specific
5	BANDGAP	Bangap
6	GND	Ground
7	INTDAC	Internal DAC

Bit 6 – RUNSTDBY Run in Standby

This bit controls the behavior of the comparator during standby sleep mode. This bit can only be written while COMPCTRLn.ENABLE is zero.

Value	Description
0	The comparator n is disabled during sleep.
1	The comparator n continues to operate during sleep.

Bits 5:4 – INTSEL[1:0] Interrupt Selection

These bits select the condition for comparator n (n=0,1) to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.

Value	Name	Description
0	TOGGLE	Interrupt on comparator output toggle
1	RISING	Interrupt on comparator output rising
2	FALLING	Interrupt on comparator output falling
3	EOC	Interrupt on end of comparison (single-shot mode only)

Bit 3 – SINGLE Single-Shot Mode

This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.

Note: To initiate a single-shot comparison the user's software must write the respective CTRLB.STARTn = 1. Be aware that If COMPCTRLn.SINGLE=1 even if comparator COMPCTRLn.ENABLE=1 the comparator is turned off to conserve power.

When CTRLB.STARTn is set by the user to start the comparison the comparator is enabled, after COMPCTRLn.SUT time has expired the comparison is made and then the comparator is again shut off to conserve power ready for the next time.

Value	Description
0	Comparator n operates in continuous measurement mode.
1	Comparator n operates in single-shot mode.

Bit 1 – ENABLE Enable

Due to synchronization, there is delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.

Writing a one to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to zero and the write is synchronized.

Value	Description
0	Writing a zero to this bit disables comparator n.
1	Writing a one to this bit enables comparator n.

39.7.13 Dac n Control

Name: DACCTRL
Offset: 0x38
Reset: 0x00000000
Property: RW

Note: This register can only be written while COMPCTRLn.ENABLE =0.

Table 39-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	SHEN1							
Access	R/W							
Reset	0							
Bit	23	22	21	20	19	18	17	16
					VALUE1[6:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	SHEN0							
Access	R/W							
Reset	0							
Bit	7	6	5	4	3	2	1	0
					VALUE0[6:0]			
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit 31 – SHEN1 DAC1 Sample and Hold Enable Operating Mode

This bit enables the odd DAC low-power operation.

Note: If DACCTRL0.SHEN0=1 user must initialize AC.CTRL0.PER and AC.CTRL0.WIDTH accordingly.

Note: This bit can be written only while COMPCTRL1.ENABLE is zero.

Value	Description
0	Continuous operation mode is enabled.
1	Sample-and-hold operation mode is enabled.

Bits 22:16 – VALUE1[6:0] DAC1 Output Value

These bits define the scaling factor for odd DAC channel voltage reference. The output voltage, V_{OUT}, is:

$$V_{OUT} = \frac{AVDD \cdot (VALUE)}{128}$$

Note: This bit can be written only while COMPCTRL1.ENABLE is zero.

Bit 15 – SHEN0 DAC0 Sample and Hold Enable Operating Mode

This bit enabled the even DAC low-power operation.

Note: If DACCTRLn.SHENn=1 user must initialize AC.CTRL.C.PER and AC.CTRL.C.WIDTH accordingly.

Note: This bit can be written only while COMPCTRL0.ENABLE is zero.

Value	Description
0	Continuous operation mode is enabled.
1	Sample-and-hold operation mode is enabled.

Bits 6:0 - VALUE0[6:0] DAC0 Output Value

These bits define the scaling factor for even DAC channel voltage reference. The output voltage, V_{OUT}, is:

$$V_{OUT} = \frac{AVDD \cdot (VALUE)}{128}$$

These bits can be written only while COMPCTRL0.ENABLE is zero.

39.7.14 Window Monitor n Control

Name: WINCTRL
Offset: 0x3C
Reset: 0x00000000
Property: RW

Table 39-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						WINTSEL[1:0]		WEN
Access						RW	RW	RW
Reset						0	0	0

Bits 2:1 – WINTSEL[1:0] Window n Interrupt Selection

These bits configure the interrupt mode for the comparator n window mode.

Value	Name	Description
0	ABOVE	Interrupt on signal above window
1	INSIDE	Interrupt on signal inside window
2	BELOW	Interrupt on signal below window
3	OUTSIDE	Interrupt on signal outside window

Bit 0 – WEN Window n Mode Enable

Note: If WINCTRL.WIN=1 then both pair of appropriate comparators must have identical COMPCTRLn.SINGLE bit value settings.

Value	Description
0	Window mode is disabled for Comparator 0 and 1
1	Window mode is enabled for Comparator 0 and 1

40. Timer/Counter for Control Applications (TCC)

40.1 Overview

The PIC32CM SG00/GC00 family of devices contain up to eight instances of the Timer/Counter for Control applications (TCC) modules: TCC0 to TCC3 are 32-bit timers and TCC4 to TCC7 are 16-bit timers.

Each TCC instance consists of a counter, a prescaler, compare/capture channels, and control logic. The counter can be set to count events or clock pulses. The counter together with the compare/capture channels can be configured to time stamp input events, allowing capture of frequency and pulse-width. It can also perform waveform generation, such as frequency generation and pulse-width modulation (PWM).

TCC contains waveform extension features for motor control, ballast control, LED control, H-bridge control, power converters, and other types of power control applications. These extensions allow for low-side and high-side output with optional dead-time insertion. Waveform extensions can also generate a synchronized bit pattern across the waveform output pins. The fault options enable fault protection for safe and deterministic handling, deactivating output signals from the microcontroller. These signals can be used to shutdown external drivers connected to the microcontroller.

Note: The number of TCC instances, TCC channel numbers and TCC extensions are device specific. The TCC configurations, such as channel numbers and features, may be reduced for some of the TCC instances.

References:

- Device Configuration Summary

40.2 Features

The following are key features of the module:

- Up to six Compare/Capture Channels (CC) with:
 - Double buffered period setting
 - Double buffered compare or capture channel
 - Circular buffer on period and compare channel registers
- Input Capture:
 - Event capture
 - Frequency capture
 - Pulse-width capture
- Waveform Generation:
 - Frequency generation
 - Single-slope pulse-width modulation (PWM)
 - Dual-slope PWM with half-cycle reload capability
- Waveform Extensions:
 - Configurable distribution of compare channels outputs across port pins
 - Low-side and high-side output with programmable dead-time insertion
 - Waveform swap option with double buffer support
 - Pattern generation with double buffer support
 - Dithering support
- Fault Protection for Safe Disabling of Drivers:

- Two recoverable fault sources
- Two non-recoverable fault sources
- Debugger can be a source of non-recoverable fault
- Input Events:
 - Two input events (TCCx_EV_0 , TCCx_EV_1) for counter
 - One input event (TCCx_MC_y) for each channel (y=0,1,2...7)
- Output Events:
 - Three output events (Count, re-trigger and overflow) are available for counter
 - One compare match/input capture event output for each channel
- Interrupts:
 - Overflow and re-trigger interrupt
 - Compare match/input capture interrupt
 - Interrupt on fault detection
- Can be used with DMA and can Trigger DMA Transactions

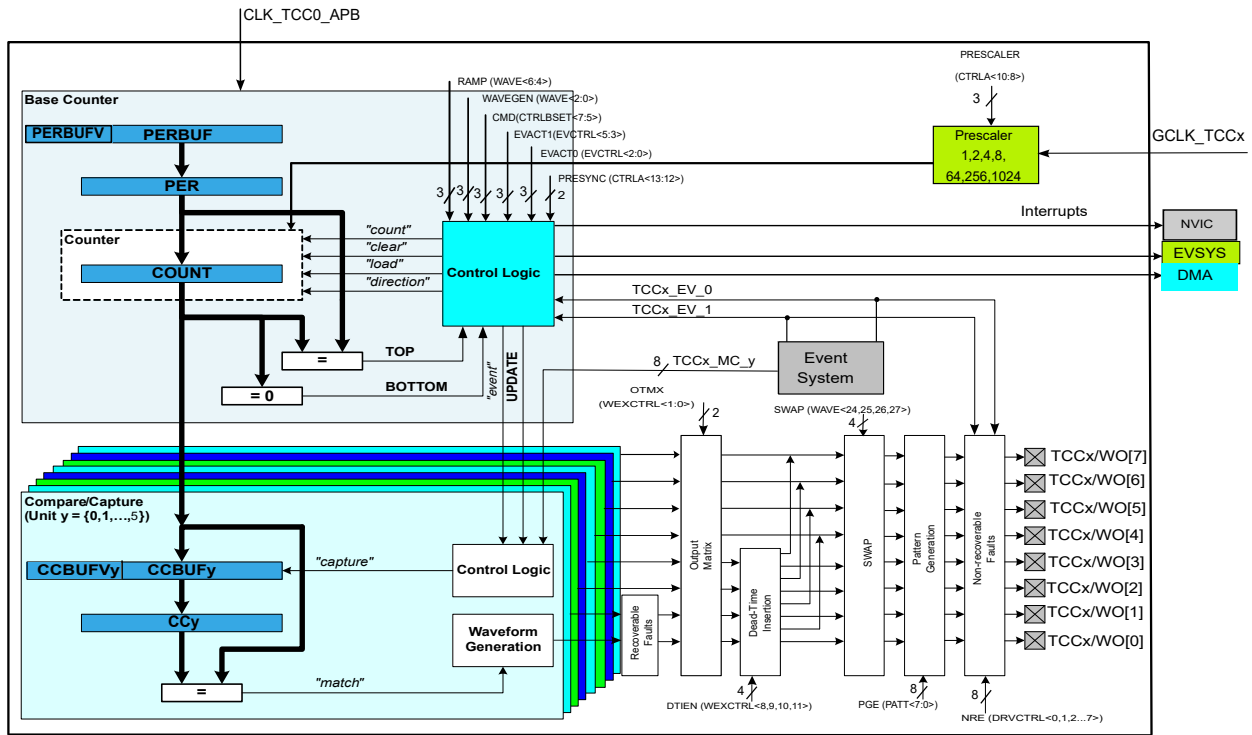
40.2.1 Feature Summary

TCC Instance	Capture-Compare Channels	Waveform Output Channel Pins	Output Channels that Support Dithering	(OTMX) Output Matrix Channels Available	(DTI) Output Channels that Support Dead Time Insertion	(SWAP) Outputs Channels that Support HS / LS swapping	(PG) Output Channels that Support Pattern Generation	Counter Size
0	6	8	1	1	1	1	1	32-bit
1	6	8	1	1	1	1	1	32-bit
2	6	8	1	1	1	1	1	32-bit
3	6	8	1	1	1	1	1	32-bit
4	2	2	0	0	0	0	0	16-bit
5	2	2	0	0	0	0	0	16-bit
6	2	2	0	0	0	0	0	16-bit
7	2	2	0	0	0	0	0	16-bit

40.3 Block Diagram

The following block diagram illustrates the Timer/Counter for Control Applications:

Figure 40-1. Timer/Counter for Control Applications Block Diagram



40.4 Signal Description

PIN NAME ^(1,2)	SIGNAL TYPE	Description
TCC0_WO0	Digital I/O	TCC0 channel 0, waveform output
TCC0_WO1	Digital I/O	TCC0 channel 1, waveform output
TCC0_WO2	Digital I/O	TCC0 channel 2, waveform output
TCC0_WO3	Digital I/O	TCC0 channel 3, waveform output
TCC0_WO4	Digital I/O	TCC0 channel 4, waveform output
TCC0_WO5	Digital I/O	TCC0 channel 5, waveform output
TCC0_WO6	Digital I/O	TCC0 channel 6, waveform output
TCC0_WO7	Digital I/O	TCC0 channel 7, waveform output
TCC1_WO0	Digital I/O	TCC1 channel 0, waveform output
TCC1_WO1	Digital I/O	TCC1 channel 1, waveform output
TCC1_WO2	Digital I/O	TCC1 channel 2, waveform output
TCC1_WO3	Digital I/O	TCC1 channel 3, waveform output
TCC1_WO4	Digital I/O	TCC1 channel 4, waveform output
TCC1_WO5	Digital I/O	TCC1 channel 5, waveform output
TCC1_WO6	Digital I/O	TCC1 channel 6, waveform output
TCC1_WO7	Digital I/O	TCC1 channel 7, waveform output
TCC2_WO0	Digital I/O	TCC2 channel 0, waveform output
TCC2_WO1	Digital I/O	TCC2 channel 1, waveform output
TCC2_WO2	Digital I/O	TCC2 channel 2, waveform output
TCC2_WO3	Digital I/O	TCC2 channel 3, waveform output
TCC2_WO4	Digital I/O	TCC2 channel 4, waveform output
TCC2_WO5	Digital I/O	TCC2 channel 5, waveform output

Signal Description (continued)		
PIN NAME ^(1,2)	SIGNAL TYPE	Description
TCC2_WO6	Digital I/O	TCC2 channel 6, waveform output
TCC2_WO7	Digital I/O	TCC2 channel 7, waveform output
TCC3_WO0	Digital I/O	TCC3 channel 0, waveform output
TCC3_WO1	Digital I/O	TCC3 channel 1, waveform output
TCC3_WO2	Digital I/O	TCC3 channel 2, waveform output
TCC3_WO3	Digital I/O	TCC3 channel 3, waveform output
TCC3_WO4	Digital I/O	TCC3 channel 4, waveform output
TCC3_WO5	Digital I/O	TCC3 channel 5, waveform output
TCC3_WO6	Digital I/O	TCC3 channel 6, waveform output
TCC3_WO7	Digital I/O	TCC3 channel 7, waveform output
TCC4_WO0	Digital I/O	TCC4 channel 0, waveform output
TCC4_WO1	Digital I/O	TCC4 channel 1, waveform output
TCC5_WO0	Digital I/O	TCC5 channel 0, waveform output
TCC5_WO1	Digital I/O	TCC5 channel 1, waveform output
TCC6_WO0	Digital I/O	TCC6 channel 0, waveform output
TCC6_WO1	Digital I/O	TCC6 channel 1, waveform output
TCC7_WO0	Digital I/O	TCC7 channel 0, waveform output
TCC7_WO1	Digital I/O	TCC7 channel 1, waveform output

Notes:

1. Each Capture Compare (CC) channel enabled with Dead Time Insertion (DTI) extension will require 2 TCC output waveforms.
2. Refer to the *Pinout* chapter for additional information on the pin mapping for this peripheral. One signal can be mapped on several pins.

40.5 Functional Description

40.5.1 Principle of Operation

The following definitions are used throughout the documentation:

Table 40-1. Timer/Counter for Control Applications - Definitions

Name	Description
TOP	The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the Waveform Generator mode in <i>Waveform Output Generation Operations</i> .
ZERO	The counter reaches ZERO when it contains all zeroes. (i.e. 0x00000000)
MAX	The counter reaches maximum when it contains all ones. (i.e. 0xFFFFFFFF)
UPDATE	The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings.
Timer	The timer/counter clock control is handled by an internal source.
Counter	The clock control is handled externally (e.g., counting external events).
CC	For compare operations, the CC are referred to as "compare channels." For capture operations, the CC are referred to as "capture channels."

There are up to six compare/capture (CC) channels starting from CC0 to CC5. The number of CC channels can be different on different instances of a TCC.

The Counter register (COUNT), Period registers with Buffer (PER and PERBUF), and Compare and Capture registers with buffers (CCy and CCBUFy) are 32-bit registers, on each TCC instance. Each Buffer register has a corresponding Buffer Valid flag in the STATUS register that indicates when the buffer contains a new value.

Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached TOP or ZERO. In either case, the TCC can generate interrupt requests, request DMA transactions, or generate events for the Event System. In Waveform Generator operation, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The Recoverable Fault event inputs are connected to the MC0 and MC1 event lines. Only asynchronous events are used internally when fault unit extension is enabled. For further details on how to configure asynchronous events routing, refer to *EVSYS – Event System*.

Recoverable fault sources can be filtered and/or windowed to avoid false triggering, for example from I/O pin glitches, by using digital filtering, input blanking, and qualification options. See also *Recoverable Faults*.

In order to support applications for different types of motors, ballasts, LEDs, H-bridges, power converters, and other types of power switching applications, the following independent units are implemented in some of the TCC instances:

- Recoverable faults and non-recoverable faults
- Output matrix
- Dead-time insertion
- Swap
- Pattern generation

See also TCC Block Diagram.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted Low Side (LS) and inverted High Side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs, and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a preconfigured value that is safe for the application. This is typically used for instant and predictable shut down and disabling high current or voltage drives. A non-recoverable fault can be recovered only by software.

The count event sources TCCx_EV_0 connected to TCE0 input and TCCx_EV_1 connected to TCE1 input are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous

action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

References:

- EVSYS

40.5.2 Basic Operation

40.5.2.1 Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel y and Compare/Capture Channel y Buffer Value registers (CCy and CCBUFy)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel y and Compare/Capture Channel y Buffer Value registers (CCy and CCBUFy)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

Reference:

- Register Synchronization

40.5.2.2 Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE=0):

- Recoverable Fault Control registers (FCTRLA and FCTRLB)
- Waveform Extension Control register (WEXCTRL)
- Drive Control register (DRVCTRL)
- Event Control register (EVCTRL)

Register Enable-protection is denoted by the 'Enable-Protected' property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE=0):

- Capture Channel x Enable bits in Control A register (CTRLA.CPTEN[31:24])
- DMA One-shot Trigger Mode (CTRLA.DMAOS)
- Full Cycle (CTRLA.FCYCLE)
- Auto Lock (CTRLA.ALOCK)
- Prescaler and Counter Synchronization Selection (CTRLA.PRESCSYNC)
- Run in Standby (CTRLA.RUNSTDBY)
- Prescaler (CTRLA.PRESCALER)
- Enhanced Resolution (CTRLA.RESOLUTION)

When CTRLA.ENABLE bit (CTRLA <1>) is '0', Enable-protected bits in the CTRLA register can be written at the same time CTRLA.ENABLE bit (CTRLA <1>) is written to '1'. However, when CTRLA.ENABLE bit (CTRLA <1>) is '1' these bits can not be written at the same time CTRLA.ENABLE bit (CTRLA <1>) is written to '0'. Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the TCC is enabled, it must be configured as outlined by the following steps:

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e., enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. If Capture mode is required, enable the channel in Capture mode by writing a '1' to the Capture Enable bit in the Control A register (i.e., for Channel 0 CTRLA.CPTEN0 bit (CTRLA <24>), for Channel 1 CTRLA.CPTEN1 bit (CTRLA <25>) etc.).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER bits (CTRLA <10:8>)).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESYNC bits (CTRLA <13:12>)).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR bit (CTRLBSET <0>)) to '1'. In this case, the COUNT register must be initialized with the desired TOP value.
4. Select the Waveform Generation operation in the WAVE register (WAVEGEN bits (WAVE <2:0>)).
5. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y=0,1,2,...7. For example, when the CC0 register is used for the duty cycle, the corresponding WAVE.POL0 bit (WAVE<16>) decides the polarity.
6. The waveform output for each individual channel can be inverted by configuring corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2,...7. For example, when CC0 register is used for the duty cycle, corresponding DRVCTRL.INVEN0 bit (DRVCTRL <16>) decides inversion.

Note: See different PWM modes for detailed information.

40.5.2.3 Enabling, Disabling, and Resetting

The TCC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TCC is disabled by writing a zero to CTRLA.ENABLE.

The TCC is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TCC, except DBGCTRL, will be reset to their initial state, and the TCC will be disabled. Refer to Control A register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

40.5.2.4 Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCCx clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

Note: When counting events, the prescaler is bypassed, the joint stream of prescaler ticks and event action ticks is called CLK_TCC_COUNT.

40.5.2.5 Counter Operation

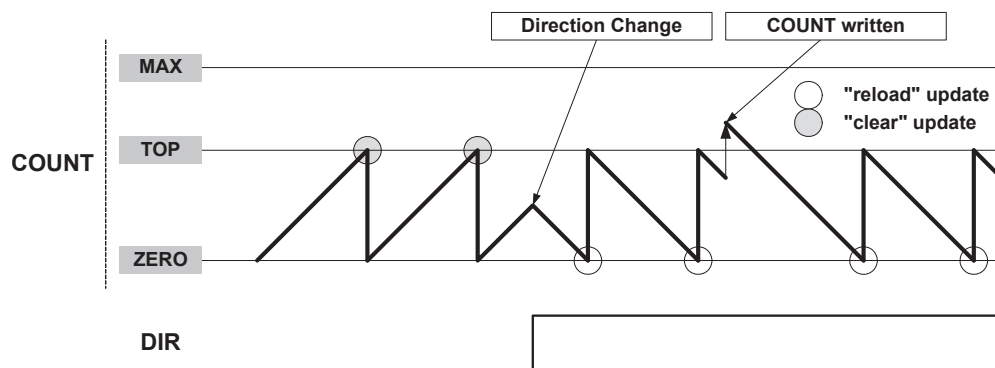
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B Set/Clear registers (CTRLBSET and CTRLBCLR). If the bit reads zero, it is counting up and if the bit reads one, it is counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero or TOP at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF bit is set. When TC is counting down, the COUNT register must be initialized to TOP value (PER or CC0 value depending on the mode).

INTFLAG.OVF bit can be used to trigger an interrupt, a DMA request or an event. An overflow/underflow occurrence (i.e. a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT). The One-Shot feature is explained in the section "Additional Features".

Figure 40-2. Counter Operation



It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running.

When starting the TCC, the COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR bit (CTRLBSET <0> or CTRLBCLR.DIR bit (CTRLBCLR <0>), unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO or TOP.

The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See the previous figure.

Stop Command

A stop command can be issued from software by using TCC Command bits in Control B Set register (CMD bits (CTRLBSET <7:5>)=0x2, STOP).

When a stop is detected while the counter is running, and if CTRLA.FCYCLE bit (CTRLA <16>) = 0 the counter stops immediately maintaining its current value, however, if CTRLA.FCYCLE bit (CTRLA <16>)=1, it waits until end of current cycle to stop on a start value.

If the waveform generation operation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State y Output Enable bit and Non-Recoverable State y Output Value bit in the Driver Control register (DRVCTRL.NREy and DRVCTRL.NRVy), and the Stop bit in the Status register is set (STATUS.STOP bit (STATUS <0>)).

Pause Event Action

A pause command can be issued when the stop event action is configured in the Input Event Action 1 bits in Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>)=0x3, STOP).

When a pause is detected, the counter will maintain its current value and all waveforms keep their current state when (CTRLA.FCYCLE (CTRLA <16>)=0), or wait to the end of the current cycle to stop on a start value and all waveforms keep their initial state when (CTRLA.FCYCLE (CTRLA <16>)= 1), until a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>) = 0x3, START).

Re-Trigger Command and Event Action

A re-trigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD (CTRLBSET <7:5>)=0x1, RETRIGGER), or from event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>)=0x1, RETRIGGER and EVCTRL.EVACT1 bits (EVCTRL <5:3>) =0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR (CTRLBSET <0>) or (CTRLBCLR.DIR (CTRLBCLR <0>)). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (TRG bit (INTFLAG <1>)). It is also possible to generate an event by writing a '1' to the Re-Trigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO bit (EVCTRL <9>)). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

Note:

When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>)=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

Start Event Action

The start action can be selected in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

Note:

When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

Count Event Action

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x5, COUNT).

Direction Event Action

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1 (EVCTRL <5:3>)=0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)) and the direction bit value is updated accordingly.

Increment Event Action

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x4, INC) and can change the Counter state when an event is received. When the TCCx_EV_0 (TCE0) event is received, the counter increments, irrespective of direction setting (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

Decrement Event Action

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1 (EVCTRL <5:3>)=0x4, DEC) and can change the Counter state when an event is received. When the TCCx_EV_1 (TCE1) event is received, the counter decrements, irrespective of direction setting (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

Non-recoverable Fault Event Action

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn=0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREy and DRVCTRL.NRVy). TCCx_EV_0 input event on TCE0 and TCCx_EV_1 input events on TCE1 must be configured as asynchronous events.

Event Action Off

If the event action is disabled (EVCTRL.EVACTn=0x0, OFF), enabling the counter will also start the counter.

40.5.2.6 Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCy) for compare operations, the counter value is continuously compared to the values in the CCy registers. This can be used for timer or for waveform operation.

The Channel y Compare/Capture Buffer Value (CCBUFy) registers provide double buffer capability. The double buffering synchronizes the update of the CCy register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD (CTRLBSET <5:3>)=0x3, UPDATE). For further details, refer to *Double Buffering*. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

40.5.2.6.1 Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a Waveform Generation Operation in the Waveform Generation Operation bits in Waveform register (WAVE.WAVEGEN (WAVE <2:0>)).
2. Optionally invert the waveform output WO[y] by writing the corresponding Waveform Output y Inversion bit in the Driver Control register (DRVCTRL.INVENy).

- Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* chapter for details.

Note: Event must not be used when the compare channel is set in waveform output operating mode.

The counter value is continuously compared with each CCy value. On a comparison match, the Match or Capture Channel y bit in the Interrupt Flag Status and Clear register (INTFLAG.MCy) will be set on the next zero-to-one transition of GCC_TCCx (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCy and/or EVCTRL.MCEOy is '1'. Both interrupt and event can be generated simultaneously. The same condition generates a DMA request.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)
- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)
- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other Waveforms Generation Operations, the update time occurs on counter wraparound, on overflow, on underflow, or on re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different Waveform Generation Operations.

Table 40-2. Counter Update and Overflow Event/interrupt Conditions

Name	Operation	TOP	Update	Output Waveform		OVF Interrupt Flag/Event	
				On Match	On Update	Up	Down
NFRQ	Normal Frequency	PER	TOP/ ZERO	Toggle	Stable	TOP	ZERO
MFRQ	Match Frequency	CC0	TOP/ ZERO	Toggle	Stable	TOP	ZERO
NPWM	Single-slope PWM	PER	TOP/ ZERO	See section 'Output Polarity' below		TOP	ZERO
DSCRITICAL	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTTOM	Dual-slope PWM	PER	ZERO			-	ZERO
DSBOTH	Dual-slope PWM	PER	TOP ⁽¹⁾ & ZERO			TOP	ZERO
DSTOP	Dual-slope PWM	PER	ZERO			TOP	-

- The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

References:

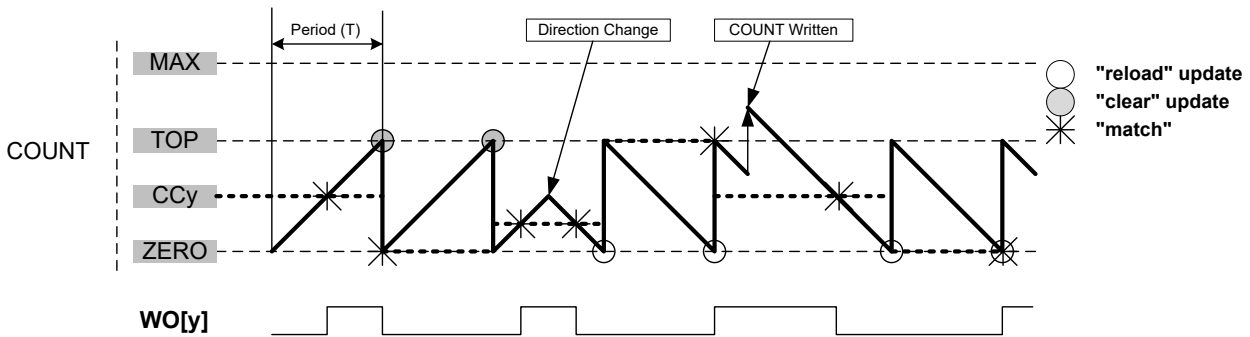
- Circular Buffer

- PORT I/O

40.5.2.6.2 Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[y]) is toggled on each compare match between COUNT and CCy, and the corresponding Match or Capture Channel y Interrupt Flag (INTFLAG.MCy) will be set.

Figure 40-3. Normal Frequency Operation



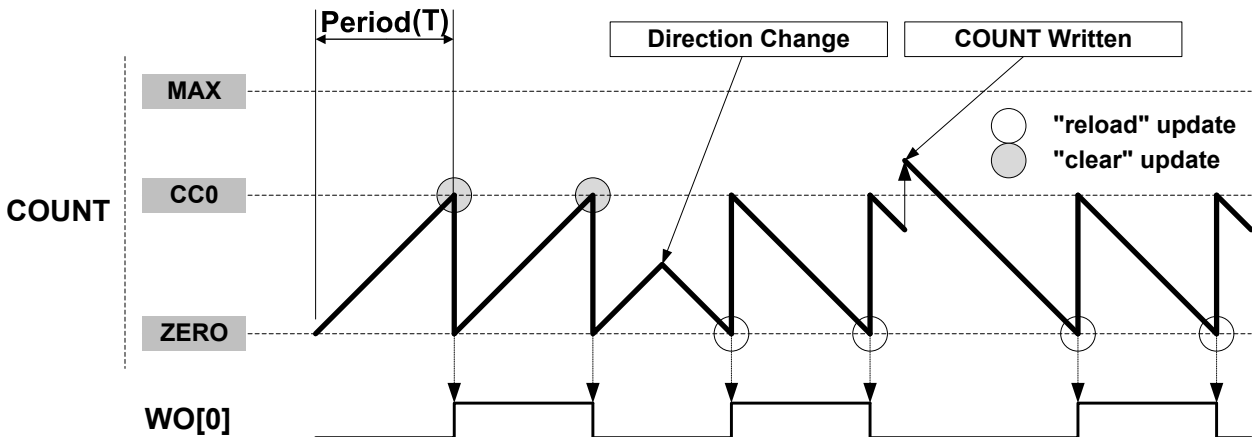
The following steps must be performed to operate the TCC in Normal Frequency (NFRQ) Waveform Generation Operations.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired Prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Select matrix routing to desired port pins for generated output waveform, by configuring OTMX bits (WEXCTRL.WEXCTRL<1:0>).
5. Set Waveform Generation Operations to Normal Frequency Operation (NRFQ) by clearing WAVE.WAVEGEN bits (WAVE<2:0> = 0)
6. Load the selected Compare/Capture (CCy) register (e.g. CC0<31:0>) with the desired compare match value. The generated output will toggle on this match.
7. Load the period register PER<31:0> with the desired time period value.
8. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> = 1)
The waveform output of a channel CCy can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
9. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling TCCx IRQ.
10. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
11. To know how to clear interrupts, see Interrupts.

40.5.2.6.3 Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

Figure 40-4. Match Frequency Operation



The following steps must be performed to operate the TCC in Match Frequency (MFRQ) Operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting CHEN bit PCHCTRL31.PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set Waveform Generation Operation to MATCH Frequency Operation (MFRQ) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 0x1).
5. Load the selected Compare/Capture (CC) register (e.g. CC0<31:0>) with the desired compare match value. The generated output will toggle on this match, PER register is not used in this operations.
6. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> = 1).
7. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
8. The waveform output can be inverted by configuring DRVCTRL.INVEN0 bit (DRVCTRL<16>).
9. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
10. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
11. To know how to clear interrupts, see Interrupts.

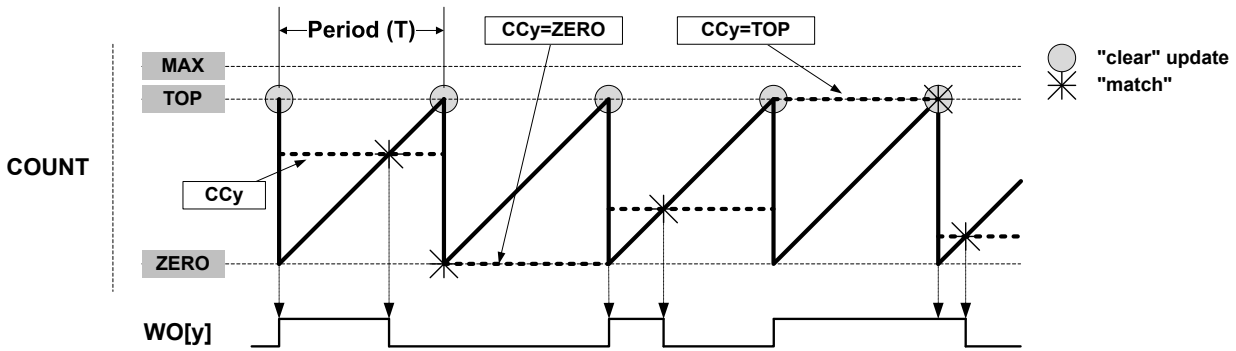
40.5.2.6.4 Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

40.5.2.6.5 Normal Pulse-Width Modulation (NPWM) Single-Slope Operation

For single-slope PWM generation, the period setting (TOP) is controlled by the PER register, and CCy controls the duty cycle of the generated waveform output. When up-counting, the WO[y] is set at start or at compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCy register values. When down-counting, the WO[y] is cleared at start or at compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCy register values.

Figure 40-5. Single-Slope PWM Operation



The following equation calculates the exact resolution for a single-slope PWM (R_{PWM_SS}) waveform:

$$R_{PWM\_SS} = \frac{\log(PER+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency ($f_{GCLK\_TCCx} = 1/GCLK\_TCCx$), and can be calculated by the following equation:

$$f_{PWM\_SS} = \frac{f_{GCLK\_TCCx}}{N(PER+1)}$$

Where N represents the prescaler divider used CTRLA.PRESCALER (CTRLA<10:8>). The selected prescaler value can be 1, 2, 4, 8, 16, 64, 256, or 1024.

The following steps must be performed to operate the TCC in single-slope PWM generation operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set Waveform Generation Operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Load the selected Compare/Capture (CCy) register (e.g. CC0<31:0>) with the desired PWM duty cycle value.
6. Load the period register PER<31:0> with the desired time period value.
7. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> = 1).
8. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
9. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y = 0,1,2,...7. For example, when the CC0 register is used for the duty cycle, the corresponding POL0 bit (WAVE<16>) decides the polarity.
10. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2,...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.

11. If overflow interrupt is used, set OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
12. Enable TCC by setting ENABLE bit (CTRLA<1>).
13. To know how to clear interrupts, see Interrupts.

40.5.2.6.6 Normal Pulse-Width Modulation (NPWM) Dual-Slope Operation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCy control the duty cycle of the generated waveform output. The figure below shows how the counter repeatedly counts from ZERO to PER and then from PER to ZERO. The waveform generator output is set on compare match when up-counting, and cleared on compare match when down-counting depends on DS mode (DSTOP, DSBOTTOM, DSBOTH or DSCRITICAL). Depending on Dual-Slope mode (DSTOP, DSBOTTOM, DSBOTH, or DSCRITICAL), an interrupt and/or event is generated on TOP (when counting upwards) and/or ZERO (when counting up or down).

In DSBOTH operation, the circular buffer must be enabled to enable the update condition on TOP.

References:

- Circular Buffer

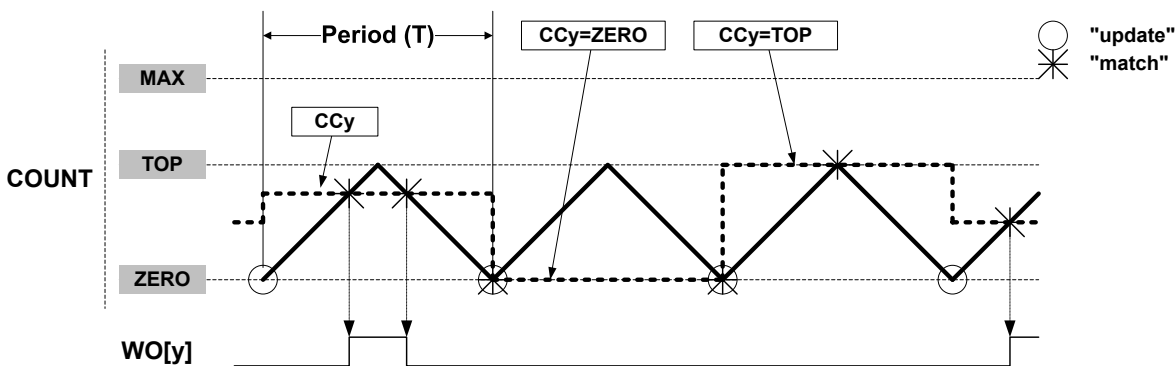
The following steps must be performed to operate the TCC in Dual-slope PWM generation operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Select prescaler Synchronization PRESCSYNC bits (CTRLA <13:12>).
5. Set waveform generation operation to Dual Slope PWM by configuring WAVE.WAVEGEN bits (WAVE<2:0>)
 - a. For DSCRITICAL operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 4)
 - b. For DSBOTTOM operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 5)
 - c. For DSBOTH operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 6)
 - d. For DSTOP operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 7)
6. Load the selected Compare/Capture (CCy) register (i.e. CC0<31:0>) with the desired PWM duty cycle value.
7. Load the period register PER<31:0> with the desired time period value.
8. Set Counter to count in up direction by clearing DIR bit (CTRLBCLR<0> = 1).
9. If DSBOTH operation (WAVEGEN bit (WAVE<2:0> = 6)) is set, enable Circular buffer by setting CIPEREN bit (WAVE<7>).
10. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
11. The dead time insertion on output matrix can be enabled by setting WEXCTRL.DTIENy bit, where y=0,1,2,3.
12. Higher side and Lower side PWM Dead-time is programmed in WEXCTRL.DTHS (WEXCTRL<30:24>) and WEXCTRL.DTLS (WEXCTRL<23:16>) respectively.

13. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y = 0,1,2,...7. For example, when the CC0 register is used for the duty cycle, the corresponding POL0 bit (WAVE<16>) decides the polarity.
14. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2,...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
15. If overflow interrupt is used, set OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
16. Enable TCC by setting ENABLE bit (CTRLA<1>).
17. For instructions on how to clear interrupts, see Interrupts.

Dual-slope Critical (DSCRITICAL) PWM operation is explained below.

Figure 40-6. Dual-Slope Pulse Width Modulation



Using dual-slope PWM results in a lower maximum operation frequency compared to single-slope PWM generation. The period (TOP) defines the PWM resolution. The minimum resolution is 1 bit (TOP=0x00000001).

The following equation calculates the exact resolution for dual-slope PWM (R_{PWM_DS}):

$$R_{PWM\_DS} = \frac{\log(PER+1)}{\log(2)}.$$

The PWM frequency f_{PWM_DS} depends on the period setting (TOP) and the peripheral clock frequency f_{GCLK_TCCx} , and can be calculated by the following equation:

$$f_{PWM\_DS} = \frac{f_{GCLK\_TCCx}}{2N \cdot PER}$$

N represents the prescaler divider used. The prescaler can be selected by CTRLA.PRESCALER bits (CTRLA<10:8>). The selected prescaler value can be 1, 2, 4, 8, 16, 64, 256, 1024.. The waveform generated will have a maximum frequency of half of the TCC clock frequency (f_{GCLK_TCCx}) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width (P_{PWM_DS}) depends on the compare channel (CCy) register value and the peripheral clock frequency (f_{GCLK_TCCx}), and can be calculated by the following equation:

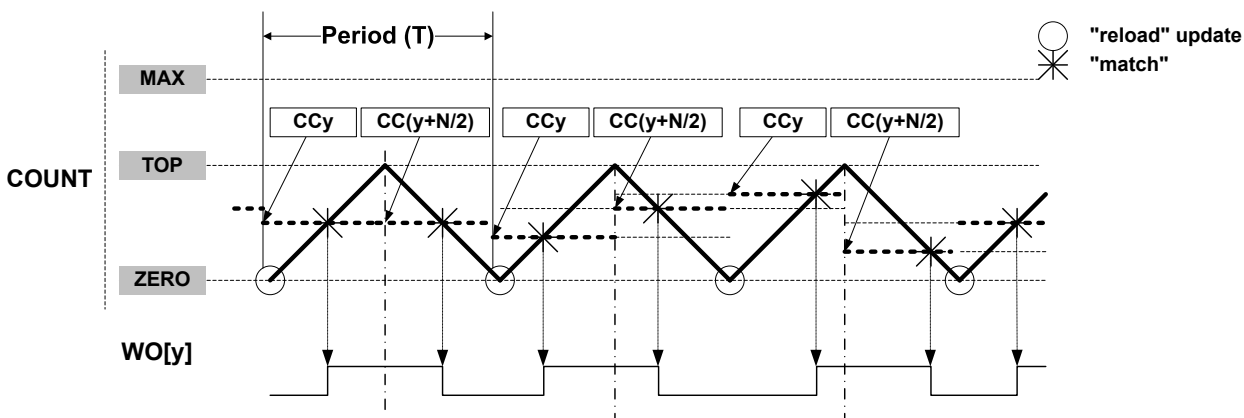
$$P_{PWM\_DS} = \frac{2N \cdot (TOP - CCy)}{f_{GCLK\_TCCx}}$$

Note: In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the MSB bit of CCy defines the Ramp on which the CCy Match interrupt or event is generated. (Rising if CCy[MSB] = 0, falling if CCy[MSB] = 1.)

40.5.2.6.7 Dual-Slope Critical PWM Generation

Critical generation operation allows generation of non-aligned centered pulses. In this operation, the period time is controlled by PER while CCy control the generated waveform output edge during up-counting and CC(y+CC_NUM/2) control the generated waveform output edge during down-counting.

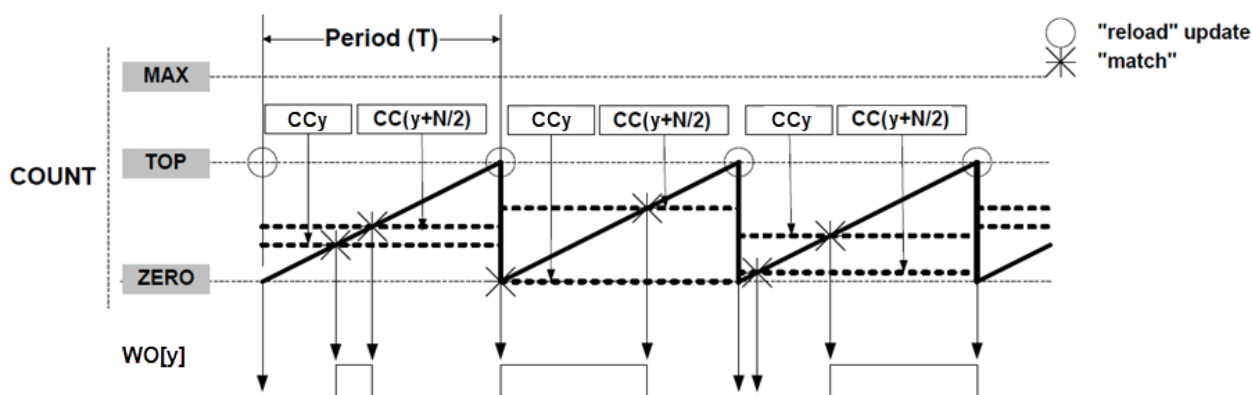
Figure 40-7. Dual-Slope Critical Pulse Width Modulation ($N=CC\_NUM$)



40.5.2.6.8 Dual-Compare PWM Generation

Dual compare PWM generation allows generation of pulses unaligned on start or end of a period. In this operation, the period time is controlled by PER, while CCy controls the waveform output leading edge and the CC(y+CC_NUM/2) controls the waveform output trailing edge.

Figure 40-8. Dual-Compare Pulse Width Modulation ($N=CC\_NUM$)



40.5.2.6.9 Output Polarity

The polarity (WAVE.POLy) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.

Table 40-3. Waveform Generation Set/Clear Conditions

Waveform Generation Operation	DIR	POL	Waveform Generation Output Update	
			Set	Clear
Single-Slope PWM	0	0	Timer/counter matches TOP	Timer/counter matches CC
		1	Timer/counter matches CC	Timer/counter matches TOP
	1	0	Timer/counter matches CC	Timer/counter matches ZERO
		1	Timer/counter matches ZERO	Timer/counter matches CC
Dual-Slope PWM	x	0	Timer/counter matches CC when counting up	Timer/counter matches CC when counting down
		1	Timer/counter matches CC when counting down	Timer/counter matches CC when counting up

In Normal and Match Frequency, the WAVE.POLy value represents the initial state of the waveform output.

40.5.2.7 Double Buffering

The Pattern (PATT), Period (PER) and Compare Channels (CCy) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV(STATUS<5>), PERBUFV(STATUS<7>) and CCBUFV) bit in the STATUS register, which indicates that the Buffer register contains a valid value that can be copied into the corresponding register. When a Buffer Valid Status flag (PATTBUFV, PERBUFV or CCBUFV) is set and the corresponding SYNCBUSY bit is set (SYNCBUSY.PATT(SYNCBUSY<5>), SYNCBUSY.PER(SYNCBUSY<7>) or SYNCBUSY.CCy), a write to the respective PATT/PATTBUF, PER/PERBUF or CCy/CCBUFy registers will generate a PAC error.

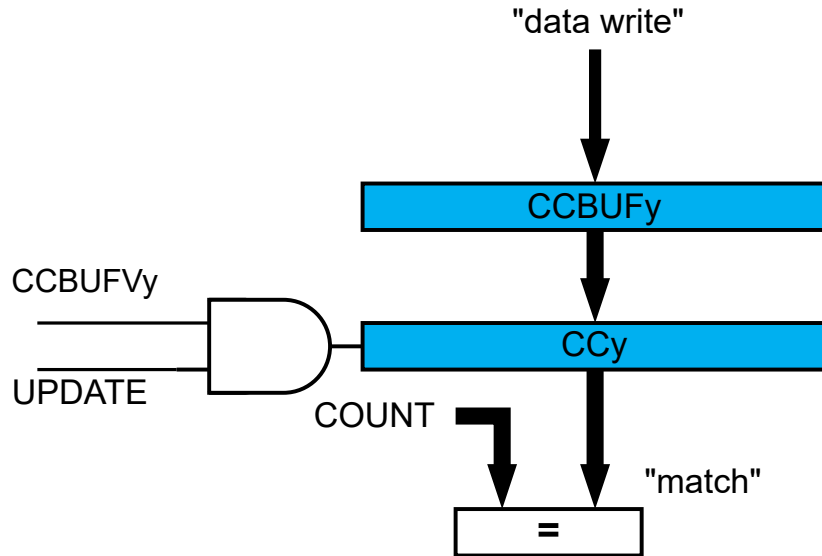
When the Buffer Valid Flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLBCLR register is cleared, (writing CTRLBCLR.LUPD(CTRLBCLR<1>) to '1'), update of register by its buffer register is allowed: Data from the buffer register will be copied into the corresponding register under hardware UPDATE conditions, then the Buffer Valid flag bit in the STATUS register is automatically cleared by the hardware.

When the buffer valid flag bit in the STATUS register is '1' (i.e. the buffer register contains a valid value), and the Lock Update bit in the CTRLB register is set to '1', (writing CTRLBSET.LUPD(CTRLBSET<1>) to '1'), update of a register by its buffer register is disabled: Data from buffer register is not copied into the corresponding register on any UPDATE conditions and the buffer valid flag bit in the STATUS register stay unchanged.

Note: Software update command (CTRLBSET.CMD(CTRLBSET<7:5>)=0x3) act independently of LUPD value.

A compare register is double buffered as in the following figure.

Figure 40-9. Compare Channel Double Buffering



The registers (PATT/PER/CCy) and corresponding Buffer registers (PATTBUF/PERBUF/CCBUFy) are available in the I/O register map, and the double buffering feature is not mandatory. Double buffering feature can be bypassed by directly writing on (PATT/PER/CCy) registers.

Changing the Period

The counter period can be changed by writing a new TOP value to the register that decides period (PER or CCy, depending on the Waveform Generation Operation), however period update on registers (PER or CCy) is effective after the synchronization delay, irrespective of whether double buffer is enabled or disabled.

Figure 40-10. Unbuffered Single-Slope Up-Counting Operation

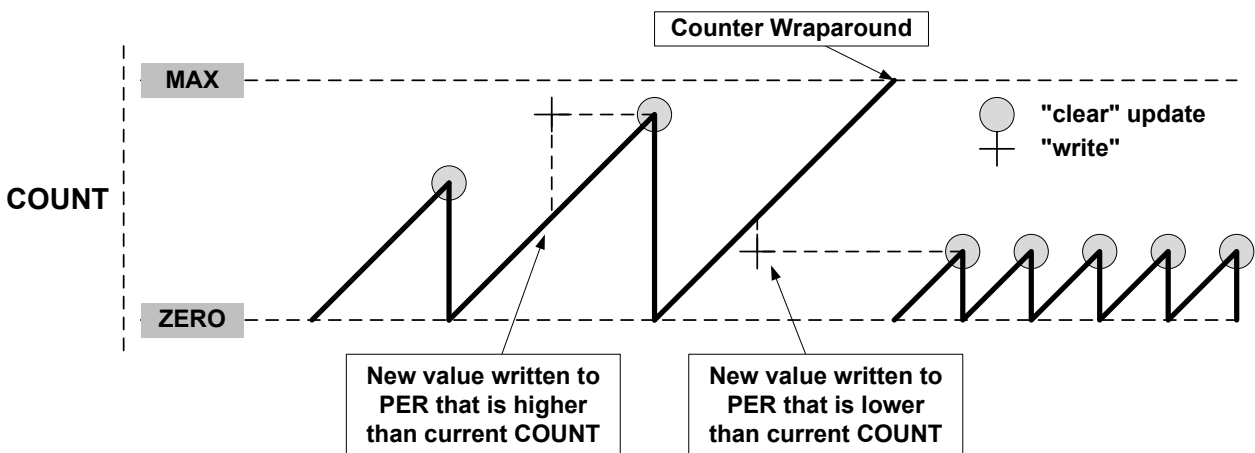
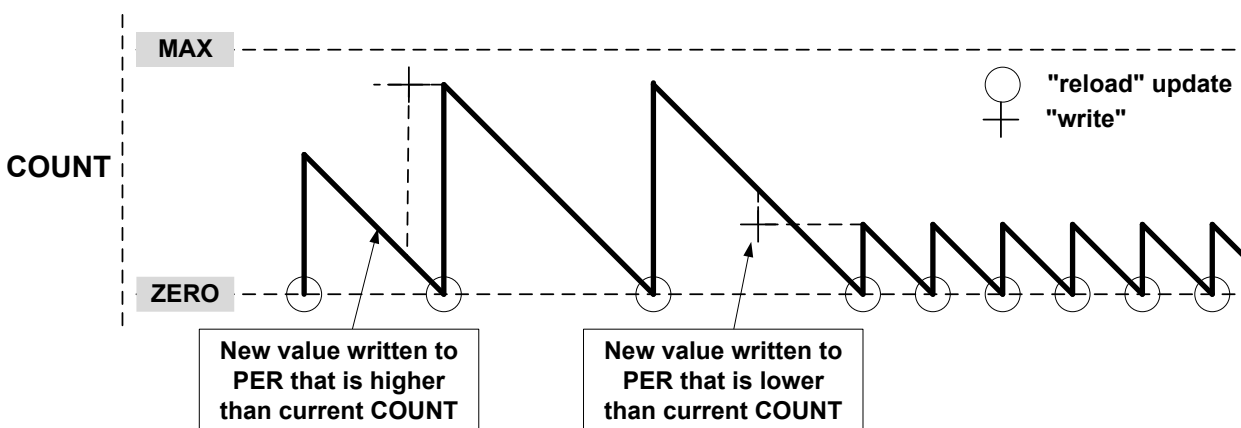
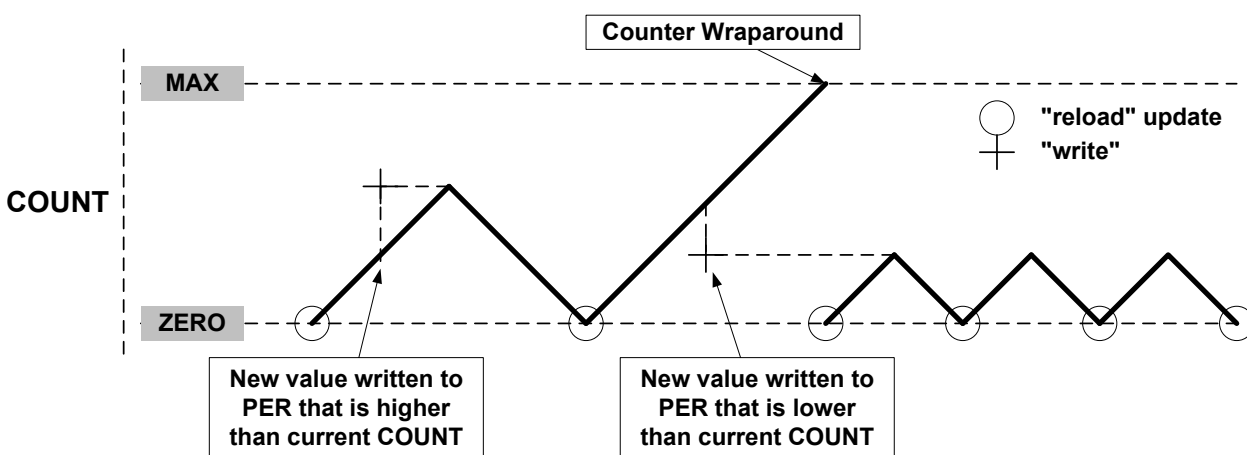
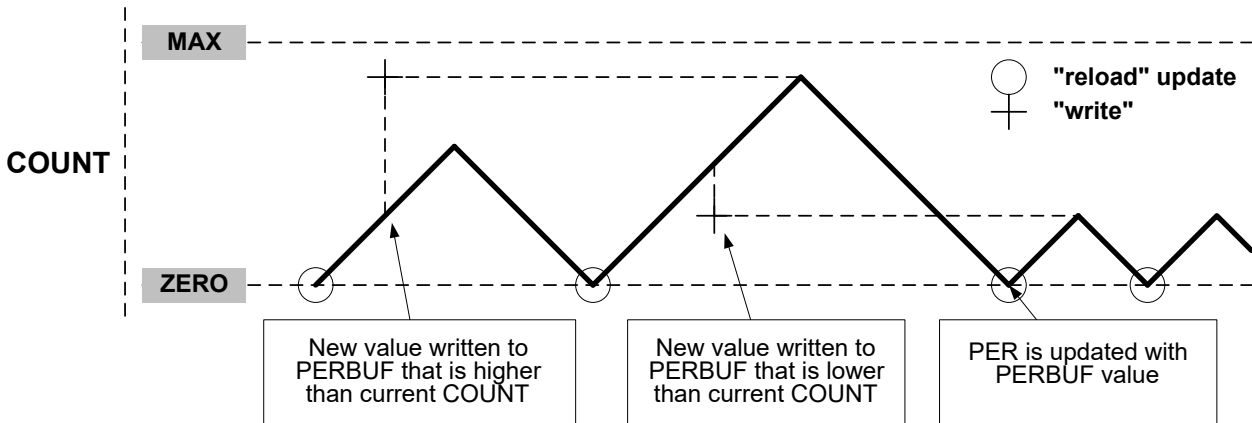


Figure 40-11. Unbuffered Single-Slope Down-Counting Operation

A counter wraparound can occur in any Waveform Generation Operation when up-counting without buffering, see the previous figure. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match. Similarly, unbuffered down-counting operation is shown in the previous figure. Unbuffered operation in dual slope mode is shown in the following figure.

Figure 40-12. Unbuffered Dual-Slope Operation

When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in *Changing the Period Using Buffer*. This prevents wraparound and the generation of odd waveforms.

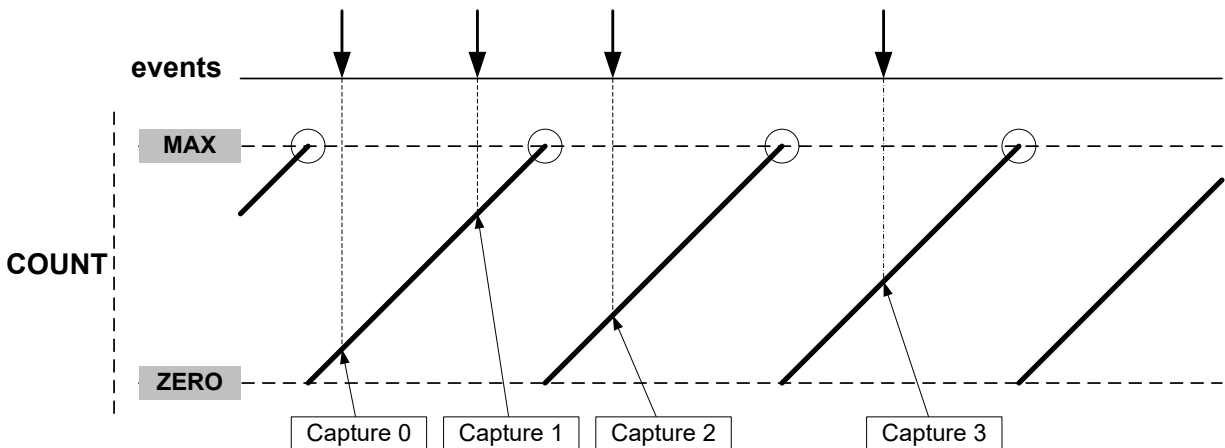
Figure 40-13. Changing the Period Using Buffering

40.5.2.8 Capture Operations

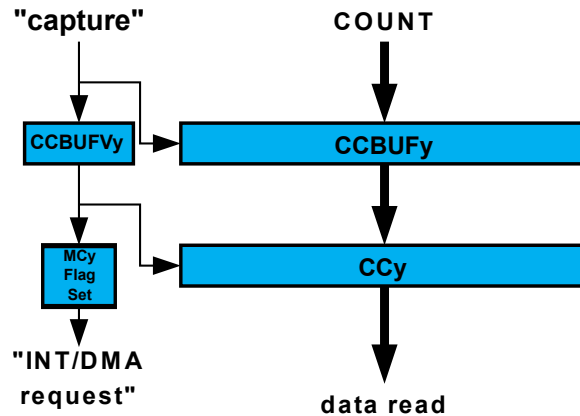
To enable and use capture operations, the Match or Capture Channel y Event Input Enable bit (MCEly) in the Event Control register must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel y Enable bit (CPTENy) in the Control A register (CTRLA.CPTENy) before capturing can be performed.

Event Capture Action

The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel. Event system channels must be configured to operate in asynchronous mode when used for capture operations.

Figure 40-14. Input Capture Timing

For input capture, the Buffer register and the corresponding CCy act like a FIFO. When CCy is empty or read, any content in CCBUFy is transferred to CCy. The Buffer Valid flag (STATUS.CCBUFV_y) is passed to set the CCy Interrupt flag (INTFLAG.MC_y) and generate the optional interrupt, event, or DMA request. The CCBUFy register value cannot be read, all captured data must be read from the CCy register.

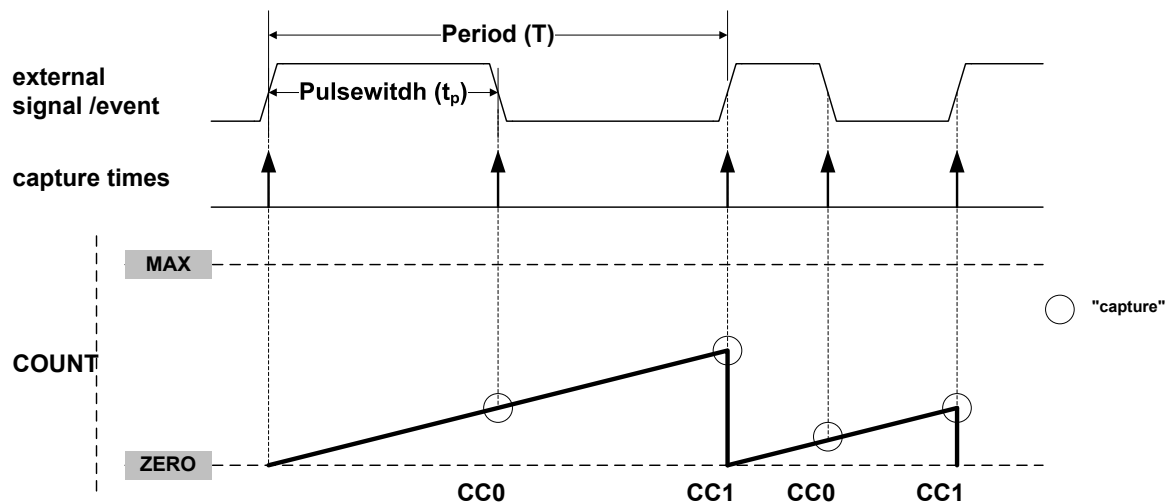
Figure 40-15. Capture Double Buffering

The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the Capture Buffer Valid flag, CCBUFV_y (For example STATUS.CCBUFV0 for CCBUF0 buffer register) is still set, the new timestamp will not be stored and INTFLAG.ERR(INTFLAG<3>) will be set.

Pulse-Width and Period (PPW) Capture Action

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency f and *dutyCycle* of an input signal, as shown below:

$$f = \frac{1}{T} \quad , \quad \text{dutyCycle} = \frac{t_p}{T}$$

Figure 40-16. PWP Capture

Selecting Pulse Width and Period (PWP) in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1(EVCTRL <5:3>)) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. In PWP event action offers, the T is captured into CC1 and t_p into CC0.

The Timer/Counter Event (TCE) n Invert Enable bit in Event Control register (EVCTRL.TCEIN_n, where $n=0,1$ and input events are TCCx_EV_0 or TCCx_EV_1) is used for event source n to select whether the reload should occur on the rising edge or the falling edge. If EVCTRL.TCEIN_n=1, the reload will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in Capture mode (CTRLA.CPTEN=1). If not, the capture action will be ignored and the channel will be enabled in compare mode of operation. When only one of these channel is required, the other channel can be used for other purposes.

The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the INTFLAG.MCy is still set, the new timestamp will not be stored and INTFLAG.ERR (INTFLAG<3>) will be set.

Note: When using PPW/PWP mode to capture the period and duty cycle of the frequency from the external pin with TCC, one must use EIC and Event System.

Note: When up-counting (CTRLBCLR.DIR(CTRLBCLR <0>)=1), counter values lower than 1 cannot be captured in Capture Minimum mode (FCTRLA.CAPTURE(FCTRLA<5:4>)=CAPTMIN or FCTRLB.CAPTURE(FCTRLA<5:4>)=CAPTMIN). To capture the full range including value 0, the TCC must be configured in Down-counting mode (CTRLBSET.DIR(CTRLBCLR <0>)=1).

40.5.3 Additional Features

40.5.3.1 One-Shot Operation

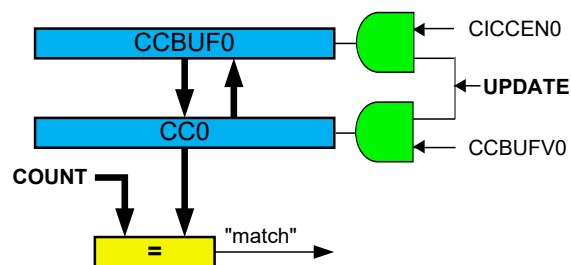
When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP(STATUS <0>)) is set and the waveform outputs are set to the value defined by DRVCTRL.NREy and DRVCTRL.NRVy.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT(CTRLBSET<2>)) and disabled by writing a '1' to CTRLBCLR.ONESHOT(CTRLBCLR<2>). When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP(STATUS <0>) is automatically cleared.

40.5.3.2 Circular Buffer

The Period register (PER) and the registers (CC0 to CC5) for Compare Channels support circular buffer operation. When circular buffer operation is enabled, the PER or CCy values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

Figure 40-17. Circular Buffer on Channel 0



40.5.3.3 Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32 or 64 fractional clock cycle base.

Dithering consists in adding some extra clock cycles on some PWM cycles, to improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Resolution bits in CTRLA register (CTRLA.RESOLUTION (CTRLA <6:5>)) and gives user three different options::

- DITH4 dithering resolution is based on the overflow of a 4 bit-counter
- DITH5 dithering resolution is based on the overflow of a 5 bit-counter
- DITH6 dithering resolution is based on the overflow of a 6 bit-counter

The least significant bits of COUNT, PER and CCy registers are used to improve accuracy of output pulse width and period. These bits are called DITHERCY bits.

The remaining bits of COUNT, PER and CCy registers define the compare value for the normal operation. The DITHERCY bits of COUNT, PER and CCy define the increment on respective dithering counter (COUNT, PER or CCy) registers to perform on each PWM cycle. If the value of dithering counter in CC register is '0' output pulse width dithering will be disabled. Similarly, if the value of dithering counter in PER register is '0' period dithering will be disabled.

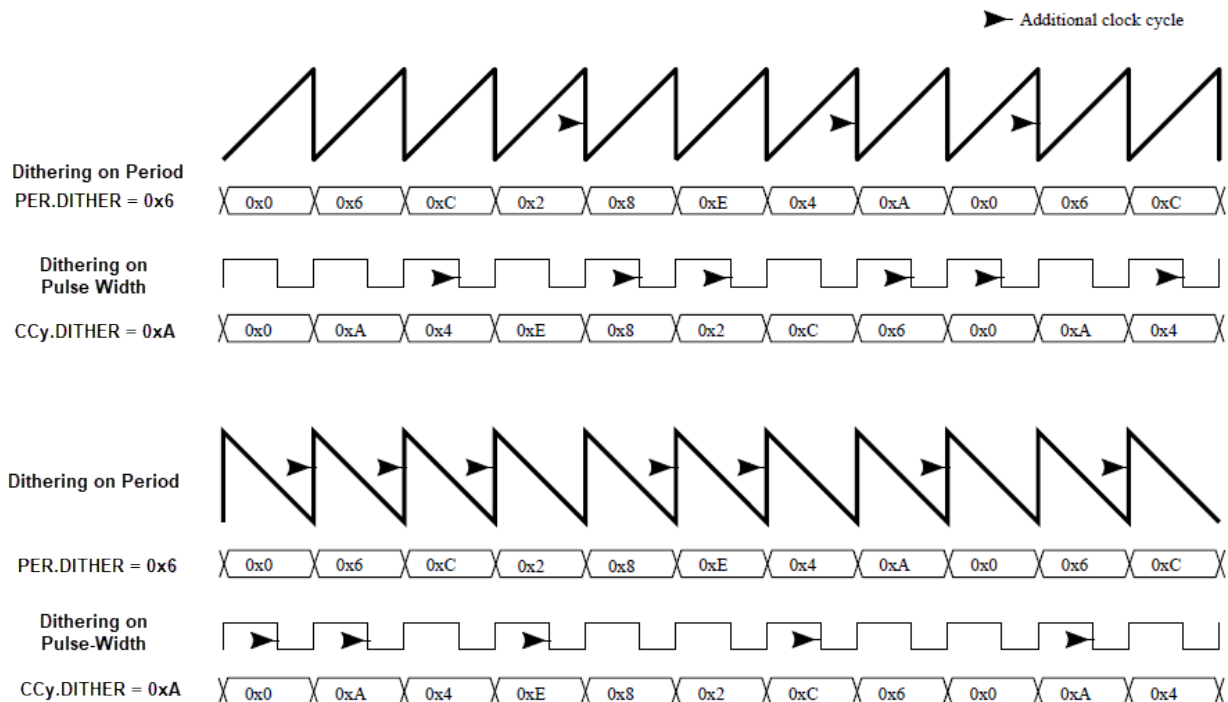
Dithering works as described below:

- In up-counting operation an extra clock cycle is inserted on each dithering counter overflow
- In down-counting operation an extra clock cycle is inserted on each PWM cycles, except when a dithering counter overflow occur

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
    int MASK;
    int value
    switch (resolution){
        DITH4: MASK = 0x0f;
        DITH5: MASK = 0x1f;
        DITH6: MASK = 0x3f;
    }
    value = cycle * dithercy;
    if ((MASK & value) + dithercy) > MASK)
        return 1;
    return 0;
}
```

Figure 40-18. Additional Clock Cycles Location in DTH4 Resolution Mode (up counting and down counting)



Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.

DITH4 mode:

If DITH4 mode is enabled for the period the 6 least significant bits from PER and COUNT registers correspond to the period DITHERCY value, rest of the bits correspond to PER or COUNT value.

$$PwmPeriod = \left(\frac{DITHERCY}{16} + PER \right) \left(\frac{1}{f_{GCLK\_TCCx}} \right)$$

DITH5 mode:

If DITH5 mode is enabled for the period the 6 least significant bits from PER and COUNT registers correspond to the DITHERCY value, rest of the bits correspond to PER or COUNT value.

$$PwmPeriod = \left(\frac{DITHERCY}{32} + PER \right) \left(\frac{1}{f_{GCLK\_TCCx}} \right)$$

DITH6 mode:

If DITH6 mode is enabled for the period the 6 least significant bits from PER and COUNT registers correspond to the DITHERCY value, rest of the bits correspond to PER or COUNT value.

$$PwmPeriod = \left(\frac{DITHERCY}{64} + PER \right) \left(\frac{1}{f_{GCLK\_TCCx}} \right)$$

Dithering on Pulse-Width

Writing DITHERCY in CCy will lead to an average PWM pulse width configured by the following formulas.

DITH4 mode:

If DITH4 mode is enabled for pulse width the 6 least significant bits from CCy and COUNT registers correspond to the output pulse width DITHERCY value, rest of the bits correspond to CCy or COUNT value.

$$PwmPulseWidth = \left(\frac{DITHERCY}{16} + CCy \right) \left(\frac{1}{f_{GCLK\_TCCx}} \right)$$

DITH5 mode:

If DITH5 mode is enabled for pulse width the 6 least significant bits from CCy and COUNT registers correspond to the output pulse width DITHERCY value, rest of the bits correspond to CCy or COUNT value.

$$PwmPulseWidth = \left(\frac{DITHERCY}{32} + CCy \right) \left(\frac{1}{f_{GCLK\_TCCx}} \right)$$

DITH6 mode:

If DITH6 mode is enabled for pulse width the 6 least significant bits from CCy and COUNT registers correspond to the output pulse width DITHERCY value, rest of the bits correspond to CCy or COUNT value.

$$PwmPulseWidth = \left(\frac{DITHERCY}{64} + CCy \right) \left(\frac{1}{f_{GCLK\_TCCx}} \right)$$

Note: The PWM period will remain static in this case.

40.5.3.4 Ramp Operations

This device supports Five Ramp Operations. All these RAMP Operations require the timer/counter running in single-slope PWM generation operation. The Ramp Operation is selected by writing to the

RAMP[2:0] bits (WAVE <6:5>) in the Waveform Control register. The Ramp Operations are broadly divided into RAMP1 Operation and RAMP2x Operation.

RAMP1 Operation

This is the default PWM operation, described in *Single-Slope PWM Generation*.

RAMP2x Operation

These operations are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see *Standard RAMP2 Operation*. These cycles are called Ramp A and Ramp B. In the Ramp A cycle, the odd channel output is disabled, and in the Ramp B cycle, the even channel output is disabled. The Ramp index changes after each update, but can be software modified using the Ramp index command bits in Control B Set register (CTRLBSET.IDXCMD(CTRLBSET <4:3>)).

The RAMP2x functionality interleaves the duty cycle of two output signals in a single TCC module. The RAMP2x functions require that the TCC module is used in single slope operation (counting up or down but not both). Each of the different RAMP2x operations use different resources available in the TCC module to achieve the interleaving output.

The TCC module is extremely flexible and can be used in many different types of applications.

Most commonly, the TCC is known for its usage in all forms of motor control applications. In motor control applications, where power is supplied by an AC source there is the need to condition power before the motor control in the system. The TCC module has built in features to enable Interleaved Power Factor Correction to reduce inefficiencies and cost in the system design for power conditioning where an electric motor is controlled. Interleaved Power Factor Correction uses two outputs of a TCC module in RAMP2x configuration to operate two parallel PFC converters at 180° out of phase from each other. It is important to maintain the phase of the two converters precisely to maintain equal load sharing between the two channels. This can all be done simply with the use of the TCC module in one of the three RAMP2x operations. Faults can be generated anywhere in the microcontroller and quickly relayed to the TCC module for immediate handling for protection or control methods.

There are four RAMP2x operations:

1. Standard RAMP2 Operation
2. Alternate RAMP2 (RAMP2A) Operation
3. Critical RAMP2 (RAMP2C) Operation
4. Critical Swapped RAMP2 (RAMP2CS) Operation

Standard RAMP2 (RAMP2) Operation

RAMP2 operation uses the PER register of the TCC module to determine the period of both output waveforms. CC0 and CC1 registers of the TCC module are used to define the duty cycle of W[0] and W[1] respectfully. To identify the cycle (A or B) of the operation, the Ramp index bit of the STATUS register STATUS.IDX (STATUS<1>) can be read.

In the following figure, the output of both WO[0] and WO[1] is going high (duty cycle begins) when the respective CC0 and CC1 value match COUNT, and goes low (duty cycle ends) when the TOP, or PER value matches the COUNT register value. This is a result of the polarity bits WAVE.POL0 (WAVE<16>) and WAVE.POL1 (WAVE<17>) being set to one. If the polarity bits were cleared, the duty cycle would begin at the reset of COUNT and conclude on the CC0 and CC1 match.

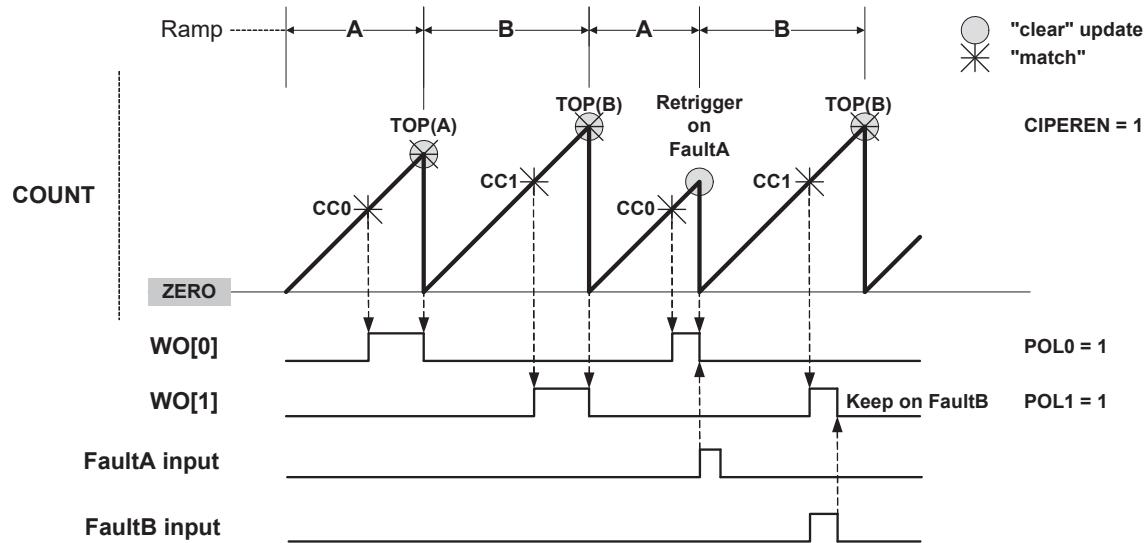
RAMP2 operation can also provide different periods for cycle A (Ramp A) and cycle B (Ramp B). To enable two different period values (TOP) for cycle A and cycle B, the period circular buffer will have to be enabled (WAVE.CIRPEREN (WAVE<7>)=1) as well as writing the PERBUF register. In this operation the following registers are used.

- PER – Period for both cycle A and cycle B when period circular buffer is not enabled

- PERBUF – At initialization, if the period circular buffer is enabled by setting WAVE.CIRPEREN (WAVE<7>) bit (WAVE<7>), then PERBUF defines cycle B period and PER defines cycle A period
- CC0 – Defines the duty cycle of cycle A (Ramp A)
- CC1 – Defines the duty cycle of cycle B (Ramp B)

In all cases updates to the period should be applied through the period buffer register, PERBUF. The use of the Ramp index bit STATUS.IDX (STATUS<1>) should be used to identify the cycle (A or B) of the operation to be sure the proper period is updated. The following steps are performed to operate the TCC in this operation and initialize the RAMP2 operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2 by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x1.
6. If cycle B requires different period value than cycle A, enable the Circular buffer by setting WAVE.CIRPEREN bit (WAVE<7>).
7. Load the selected Compare/Capture (CC0) register with the desired PWM duty cycle value for Cycle A.
8. Load the another Compare/Capture (CC1) register with the desired PWM duty cycle value for Cycle B.
9. Load the period register PER<31:0> with the desired time period value (TOP) for cycle A.
10. If the circular buffer is enabled, load the period register PERBUF<31:0> with the desired time period value (TOP) for cycle B.
11. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
12. In this operation the output polarity of a selected output signal (if needed) should be set as follow
 - a. Set WAVE.POL0 bit (WAVE<16>) to invert WO[0] output.
 - b. Set WAVE.POL1 bit (WAVE<17>) to invert WO[1] output
13. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if compare match interrupt is used, set INTENSET.MC0 bit (INTENSET<16> and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
14. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
15. For information on how to clear interrupts, see Interrupts.

Figure 40-19. RAMP2 Standard Operation

Alternate RAMP2 (RAMP2A) Operation

In RAMP2A operation, CC0 is required to use the circular buffer to provide two different duty cycles for cycle A (Ramp A) and cycle B (Ramp B). Updates to the duty cycle can be made by writing the CCBUF0 register at the time of an INTFLAG.OVF (INTFLAG<0>) or INTFLAG.MCy flags flag or MC flag. Writing CCBUF0 at the time of an INTFLAG.OVF (INTFLAG<0>) flag will change the duty-cycle of cycle B. Writing CCBUF0 at the time of an INTFLAG.MCy flag will change the duty cycle of cycle A. To further identify the cycle (A or B) of the operation refer to the Ramp index bit of the STATUS register, STATUS.IDX bit (STATUS<1>). To eliminate the need of monitoring the INTFLAG.OVF (INTFLAG<0>) or INTFLAG.MCy flags for the update to the duty-cycle, the DMA can be used to populate the appropriate buffer register.

RAMP2A operation uses the PER register of the TCC module to determine the period of both output waveforms. If different periods are needed for cycle A and cycle B outputs, the period circular buffer (CIPEREN) will have to be enabled. As mentioned for the duty cycle, the PER and PERBUF registers can be treated in the same manner as the CC0 and CCBUF0 to provide the period updates at the appropriate times. If the DMA is to be used, a separate DMA channel will need to be initialized for the period management.

- **PER** – Defines the period for both cycle A and cycle B if the period circular buffer is NOT enabled
- **PERBUF** – At startup PERBUF defines cycle B period and PER defines cycle A period. After the TCC module is started PERBUF holds the period value (TOP) of the next cycle
- **CC0** – Defines the duty cycle for both cycle A and cycle B if the CC circular buffer is NOT enabled
- **CCBUF0** – At startup CCBUF0 defines cycle B duty cycle and CC0 defines cycle A duty cycle. After the TCC module is started CCBUF holds the duty cycle value of the next cycle

In all cases for use with or without DMA in RAMP2A operation, updates to the period or duty cycle should be applied through the buffer registers (PERBUF or CCBUFy). If both the period and duty cycle have different values in cycle A and cycle B and the DMA is used, two DMA channels will be needed, one for PERBUF and one for CCBUFy. Refer to *RAMP2 Alternate Operation*.

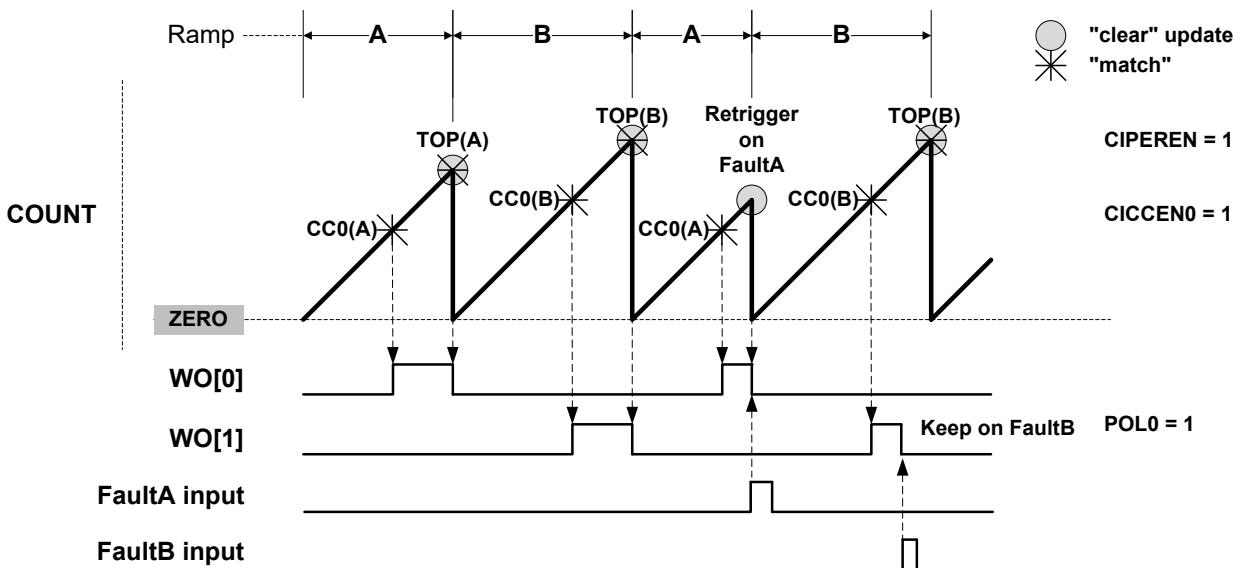
The following steps are performed to operate the TCC in this mode.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).

2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2A by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x2.
6. If cycle A requires different period value than cycle B, enable the Circular buffer for period by setting WAVE.CIPEREN bit (WAVE<7>).
7. Load Compare/Capture (CC0<31:0>) register with the desired PWM duty cycle value for Cycle A.
8. Load Compare/Capture buffer (CCBUF0<31:0>) register with the desired PWM duty cycle value for Cycle B.
9. Load the period register PER<31:0> with the desired time period value (TOP) for cycle A.
10. Load the period register PERBUF<31:0> with the desired time period value(TOP) for cycle B.
11. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
12. In this operation the output polarity of both output signals on WO[0] and WO[1], is controlled by setting WAVE.POL0 bit (WAVE<16>)
13. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if the compare match interrupt is used, set INTENSET.MC0 bit (INTENSET<16>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
14. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
15. For information on how to clear interrupts, see Interrupts.

Note: Retrigger in RAMP2 operations (RAMP2, RAMP2A, RAMP2C, RAMP2CS) is not supported if a prescaler is used (CTRLA.PRESCALER != 0) and the retrigger of the counter is done on the next GCLK (CTRLA.PRESCSYNC = GCLK or CTRLA.PRESCSYNC = RESYNC). If use of the prescaler is required, configure the retrigger of the counter on the next prescaler clock (CTRLA.PRESCSYNC = PRESC).

Figure 40-20. RAMP2 Alternate Operation



Critical RAMP2 (RAMP2C) Operation

Critical RAMP2 operation provides a way to cover RAMP2 operation requirements without the update constraint associated with the use of circular buffers. In this operation, CC0 is controlling the period of Ramp A and PER is controlling the period of Ramp B. When using more than two channels, WO[0] output is controlled by CC2 (HIGH) and CC0 (LOW). On TCC with 2 channels, a pulse on WO[0] will last the entire period of Ramp A, if WAVE.POL0 bit (WAVE<16>)=0. Refer to *RAMP2 Critical Operation With More Than 2 Channels* for more information.

The RAMP2C operation is the simplest to setup and understand as circular buffers are not used in the implementation. The period and duty cycle of output is controlled as shown below.

- PER – Defines the period for cycle B (Ramp B)
- CC0 – Defines the period for cycle A (Ramp A)
- CC1 – Defines the duty cycle for cycle B (Ramp B)
- CC2 – Defines the duty cycle for cycle A (Ramp A)

RAMP2 Critical Operation With More Than 2 Channels illustrates the RAMP2CS operation with three active channels. The following steps are performed to operate the TCC in this operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2C by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x3.
6. Load Compare/Capture (CC1<31:0>) register with the desired PWM duty cycle value for Cycle B.
7. Load Compare/Capture buffer register (CC2<31:0>) register with the desired PWM duty cycle value for Cycle A.
8. Load the period register PER<31:0> with the desired time period value (TOP) for cycle B.
9. Load the Compare/Capture register (CC0<31:0>) with the desired time period value (TOP) for cycle A.
10. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
11. In this operation the output polarity of a selected output signal should be set as follow
 - a. Set WAVE.POL2 bit (WAVE<18>) as this controls WO[0] output.
 - b. Set WAVE.POL1 bit (WAVE<17>) as this controls WO[1] output
12. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) ,if compare match interrupt is used, set INTENSET.MC2 bit (INTENSET<18>) for Ramp A match, INTENSET.MC1 bit (INTENSET<17>) for Ramp B match interrupt. Configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
13. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
14. For information on how to clear interrupts, see Interrupts.

Figure 40-21. RAMP2 Critical Operation With More Than 2 Channels

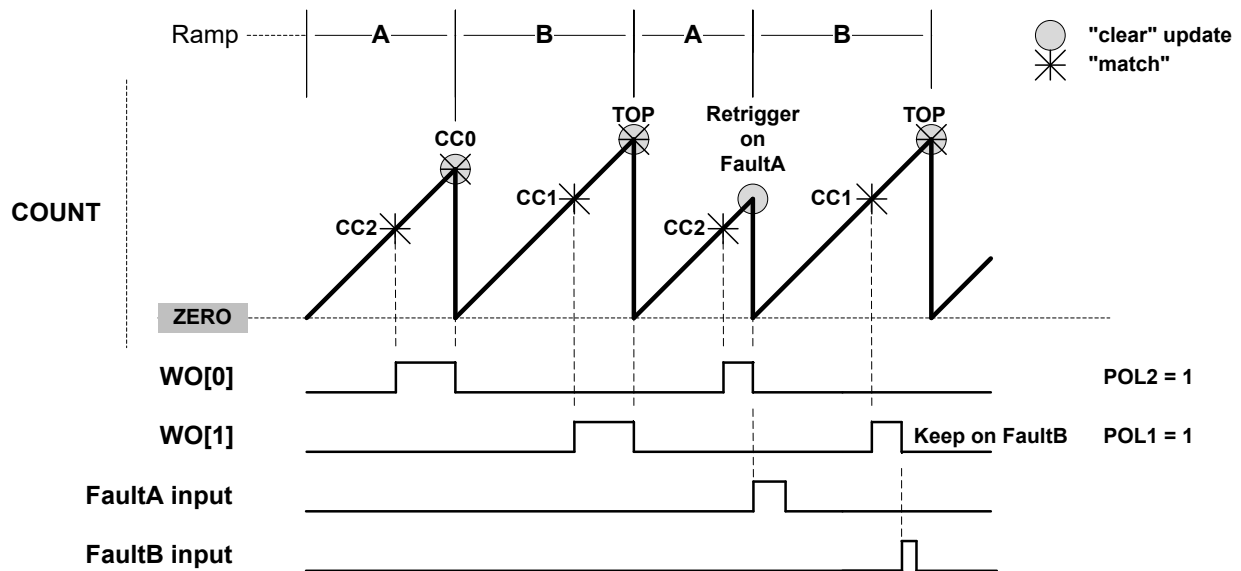
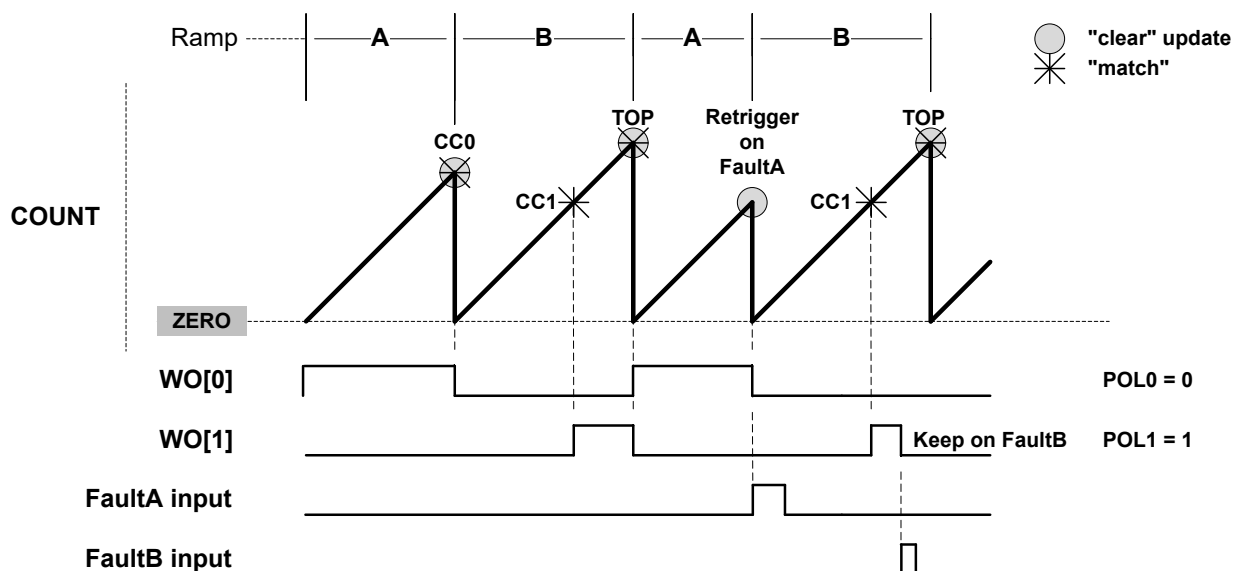


Figure 40-22. RAMP2 Critical Operation With 2 Channels



Critical Swapped RAMP2 (RAMP2C) Operation

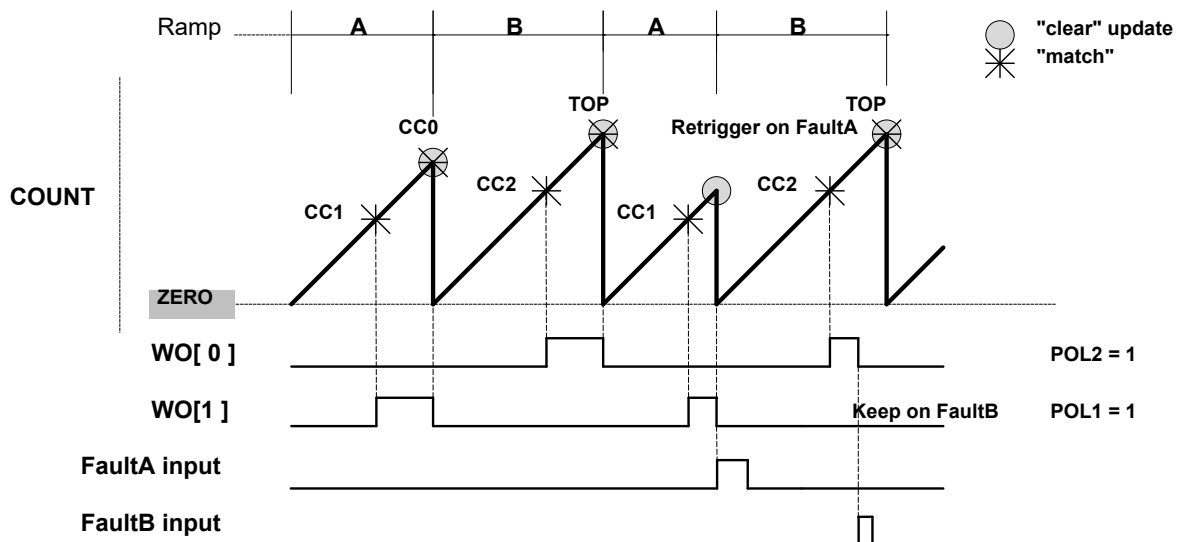
In RAMP2CS variant, WO[0] and WO[1] active ramp are inverted, WO[0] is active on RAMPB and WO[1] is active on RAMPA. The RAMP2CS operation is similar to RAMP2C operation and does not use circular buffers in the implementation. The period and duty cycle of output is controlled as shown below.

- PER – Defines the period for cycle B (Ramp B)
- CC0 – Defines the period for cycle A (Ramp A)
- CC1 – Defines the duty cycle for cycle A (Ramp A)
- CC2 – Defines the duty cycle for cycle B (Ramp B)

RAMP2 Critical Swap Operation illustrates the RAMP2CS operation. The following steps are performed to operate the TCC in this operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2CS by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x4.
6. Load Compare/Capture (CC1<31:0>) register with the desired PWM duty cycle value for Cycle A (Ramp A).
7. Load Compare/Capture buffer register (CC2<31:0>) register with the desired PWM duty cycle value for Cycle B (Ramp B).
8. Load the period register PER<31:0> with the desired time period value (TOP) for cycle B.
9. Load the Compare/Capture register (CC0<31:0>) with the desired time period value (TOP) for cycle A.
10. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
11. In this operation the output polarity of a selected output signal should be set as follows:
 - a. Set WAVE.POL2 bit (WAVE<18>) as this controls WO[0] output.
 - b. Set WAVE.POL1 bit (WAVE<17>) as this controls WO[1] output
12. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if compare match interrupt is used, set INTENSET.MC1 bit (INTENSET<17>) for Ramp A match, INTENSET.MC2 bit (INTENSET<18>) for Ramp B match interrupt. Configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
13. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
14. For information on how to clear interrupts, see Interrupts.

Figure 40-23. RAMP2 Critical Swap Operation



40.5.3.5 Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to inactive state either as long as the fault condition is present, or from the first valid fault condition detection on until the end of the timer/counter cycle.

Fault Inputs

The first two channel input events (TCCx_MC_0 and TCCx_MC_1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM operation.

Fault Filtering

There are three filters available for both Fault A and Fault B input. They are configured by the corresponding Recoverable Fault Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

Input Filtering When disabled (FCTRLx.FILTERVAL = 0), the action of a fault on the output port is asynchronous. This helps the fault action to be performed on the output of the compare channel, even when the fault is caused by the loss of the system clock. Therefore, when a fault occurs, the system will immediately and asynchronously disable the compare channel output as long as a fault is present. To avoid TCC control block corruption by a glitch on a fault input line, the fault action on the TCC control block (retrigger, capture, etc..) is synchronized on internal clock domain. A digital filter can be enabled and configured by the Fault Filter Value bits in the Fault Configuration registers (FCTRLA.FILTERVAL (FCTRLA <28:24>) and FCTRLB.FILTERVAL (FCTRLB <28:24>)). If the fault width is less than the FILTERVAL (in TCC clock cycles), will be ignored. A valid fault event action on the TCC FSM, will be then delayed by the clock cycles represented by the FILTERVAL.

Fault Blanking This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge triggering configuration to the Fault A or Fault B Blanking Mode bits in the Recoverable Fault Configuration registers (FCTRLA.BLANK (FCTRLA <6:5>) or FCTRLB.BLANK (FCTRLB <6:5>)). The desired duration of the blanking must be written to the Fault Blanking Time bits (FCTRLA.BLANKVAL (FCTRLA <23:16>) or FCTRLB.BLANKVAL (FCTRLB <23:16>)). The blanking time t_b is calculated by

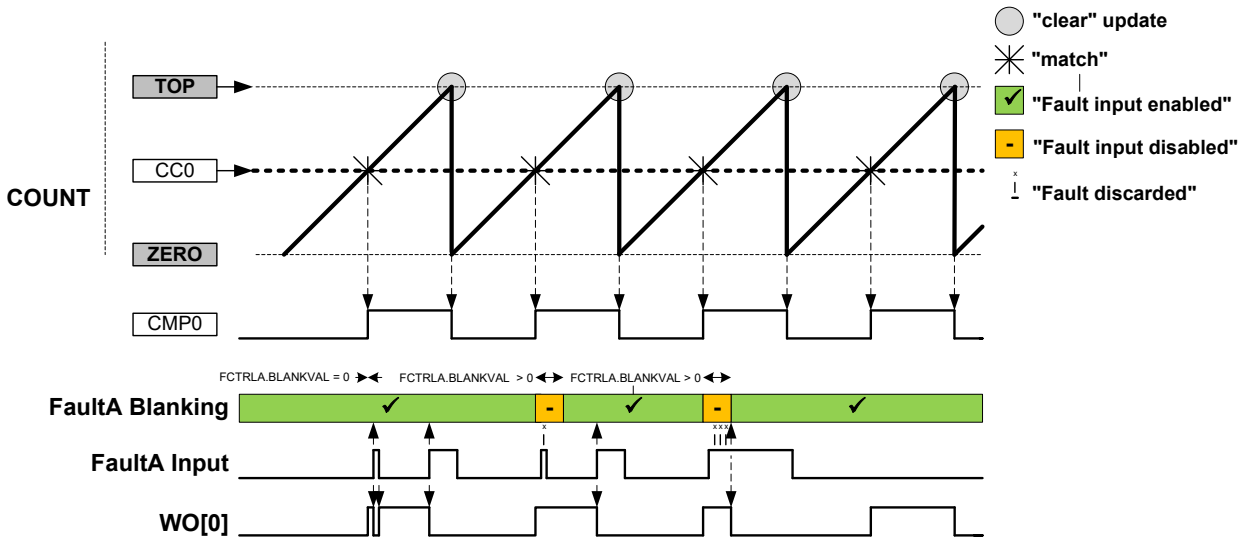
$$t_b = \frac{1 + \text{BLANKVAL}}{f_{\text{GCLK\_TCCx\_PRESC}}}$$

Here, $f_{\text{GCLK\_TCCx\_PRESC}}$ is the frequency of the prescaled peripheral clock frequency $f_{\text{GCLK\_TCCx}}$.

The prescaler is enabled by writing '1' to the Fault A or Fault B Blanking Prescaler bit (FCTRLA.BLANKPRESC (FCTRLA <15>) or FCTRLB.BLANKPRESC (FCTRLB <15>)). When disabled, $f_{\text{GCLK\_TCCx\_PRESC}} = f_{\text{GCLK\_TCCx}}$. When enabled, $f_{\text{GCLK\_TCCx\_PRESC}} = f_{\text{GCLK\_TCCx}}/64$.

The maximum blanking time (FCTRLA.BLANKVAL (FCTRLA <23:16>) or FCTRLB.BLANKVAL (FCTRLB <23:16>))=

255) at $f_{\text{GCLK\_TCCx}}=96\text{MHz}$ is $2.67\mu\text{s}$ (no prescaler) or $170\mu\text{s}$ (prescaling). For $f_{\text{GCLK\_TCCx}}=1\text{MHz}$, the maximum blanking time is either $170\mu\text{s}$ (no prescaling) or 10.9ms (prescaling enabled).

Figure 40-24. Fault Blanking in RAMP1 Operation with Inverted Polarity**Fault Qualification**

This is enabled by writing a '1' to the Fault A or Fault B Qualification bit in the Recoverable Fault A or Fault B Configuration register ((FCTRLA.QUAL(FCTRLA <4>) or FCTRLB.QUAL(FCTRLB <4>)). When the recoverable fault qualification is enabled (FCTRLA.QUAL(FCTRLA <4>)=1 or FCTRLB.QUAL(FCTRLB <4>)=1, the fault input is disabled all the time and the corresponding channel output has an inactive level, as shown in the figures below.

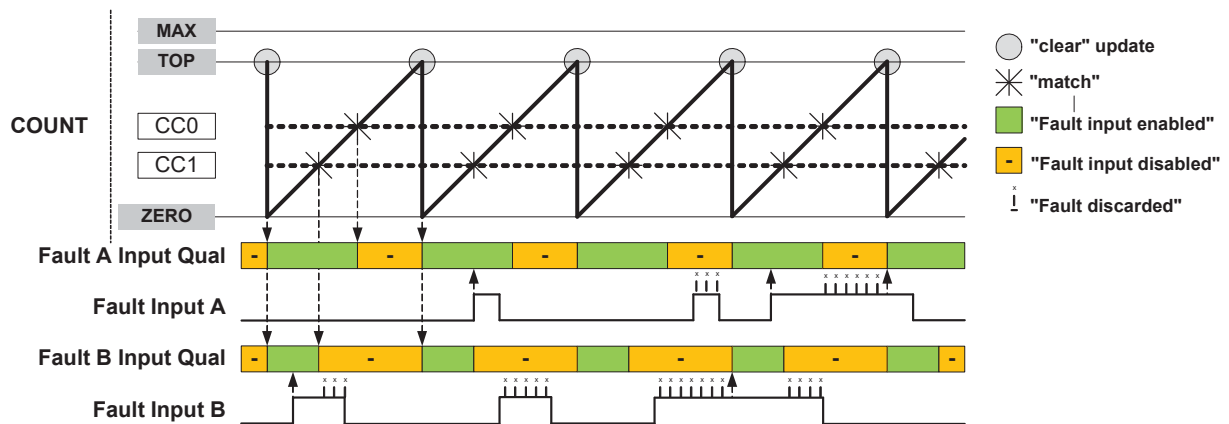
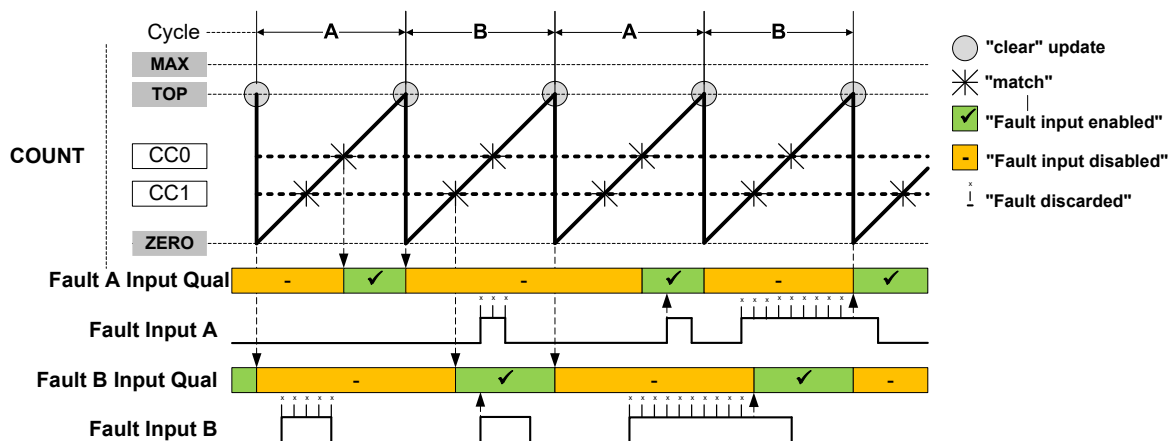
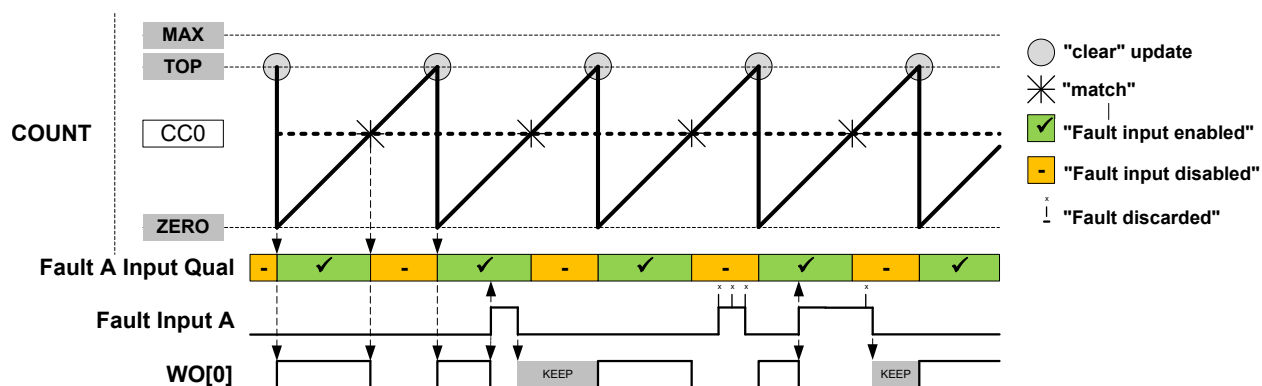
Figure 40-25. Fault Qualification in RAMP1 Operation

Figure 40-26. Fault Qualification in RAMP2 Operation with Inverted Polarity

Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; therefore two or more actions can be enabled at the same time to achieve a result that is a combination of fault actions.

Keep Action This is enabled by writing the Fault A or Fault B Keeper bit in the corresponding Recoverable Fault A or Fault B Configuration register (FCTRLA.KEEP(FCTRLA <3>) or FCTRLB.KEEP(FCTRLB <3>) to '1'. When enabled, the corresponding channel output will be clamped to zero as long as the fault condition is present. The clamp will be released on the start of the first cycle after the fault condition is no longer present, see next Figure.

Figure 40-27. Waveform Generation with Fault Qualification and Keep Action

Restart Action This is enabled by writing the Fault A or Fault B Restart bit in corresponding Recoverable Fault A or Fault B Configuration register (FCTRLA.RESTART(FCTRLA <7>) = 1 or FCTRLB.RESTART(FCTRLB <7>) to '1'. When enabled, the timer/counter will be restarted as soon as the corresponding fault condition is present. The ongoing cycle is stopped and the timer/counter starts a new cycle, see [Waveform Generation in RAMP1 Operation with Restart Action](#). In RAMP1 operation, when the new cycle starts, the compare outputs will be clamped to inactive level as long as the fault condition is present.

Note: For RAMP2 operation, when a new timer/counter cycle starts the cycle index will change automatically, see [Waveform Generation in RAMP2 Operation with Restart Action](#). Fault A and Fault B are qualified only during the cycle A and cycle B respectively: Fault A is disabled during cycle B, and Fault B is disabled during cycle A.

Figure 40-28. Waveform Generation in RAMP1 Operation with Restart Action

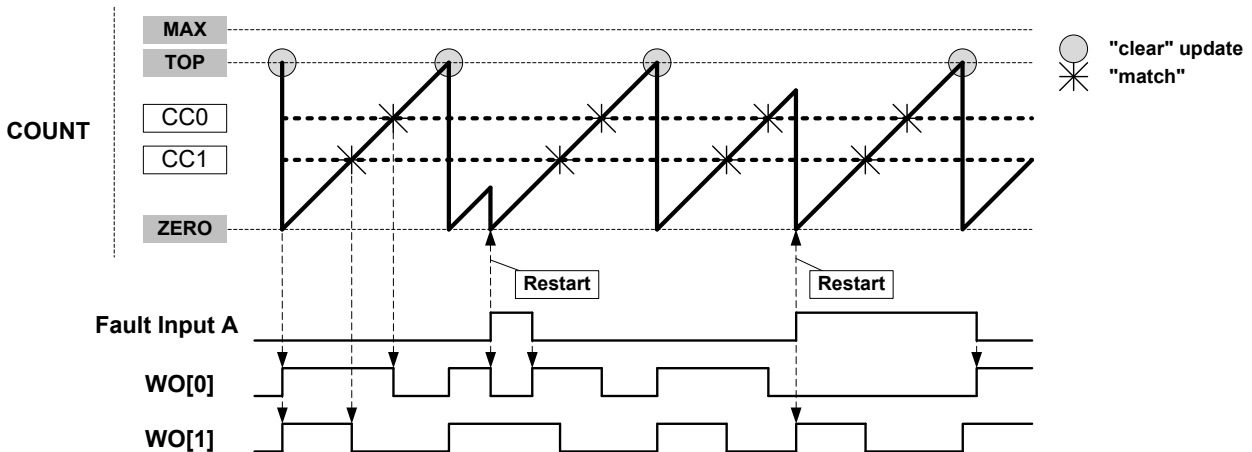
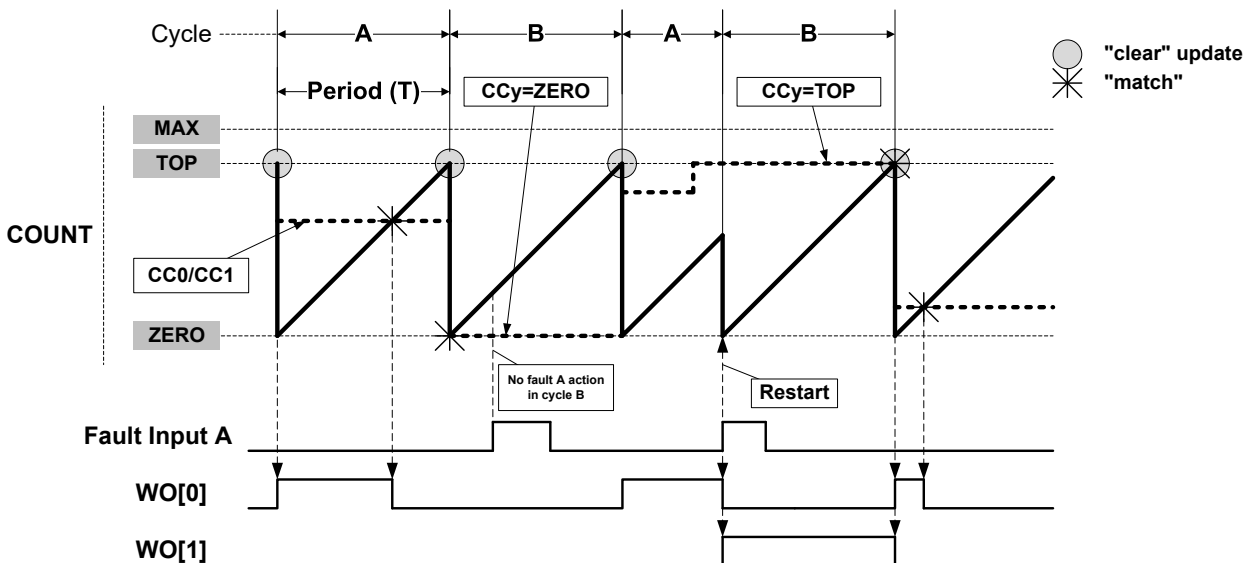


Figure 40-29. Waveform Generation in RAMP2 Operation with Restart Action



Capture Action

Several capture actions can be selected by writing the Fault A or Fault B Capture Action bits in the Fault A or Fault B Control register (FCTRLA.CAPTURE(FCTRLA <14:12>) or FCTRLB.CAPTURE(FCTRLB <14:12>). When one of the capture operations is selected, the counter value is captured when the fault occurs. The following capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details refer to *Capture Operations*
- CAPTMIN - gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued.
- CAPTMAX - gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued, see *Capture Action "CAPTMAX"* in the FCTRLA or FCTRLB register.
- LOCMIN - notifies by event or interrupt when a local minimum captured value is detected.
- LOCMAX - notifies by event or interrupt when a local maximum captured value is detected.

- DERIVO - notifies by event or interrupt when a local extreme captured value is detected, see *Capture Action "DERIVO"* in the FCTRLA or FCTRLB register.

CCy Content:

In CAPTMIN and CAPTMAX operations, CCy keeps the respective extreme captured values, see *Capture Action "CAPTMAX"*. In LOCMIN, LOCMAx or DERIVO operation, CCy follows the counter value at fault time, see *Capture Action "DERIVO"*.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCy register value to a value different from zero (for CAPTMIN) and TOP (for CAPTMAX). If the initial value of the CCy register is zero for CAPTMIN and TOP for CAPTMAX, no captures will be performed using the corresponding channel.

MCy Behaviour:

In LOCMIN and LOCMAx operation, capture is performed on each capture event. The MCy interrupt flag is set only when the captured value is above or equal for LOCMIN and below or equal for LOCMAx to the previous captured value. So interrupt flag is set when a new relative local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected. DERIVO is equivalent to an OR function of (LOCMIN, LOCMAx).

In CAPT operation, capture is performed on each capture event. The MCy interrupt flag is set on each new capture.

In CAPTMIN and CAPTMAX operation, capture is performed only when on capture event time, the counter value is lower (for CAPTMIN) or higher (for CAPTMAX) than the last captured value. The MCy interrupt flag is set only when on capture event time, the counter value is greater than or equal (for CAPTMIN), or less than or equal (for CAPTMAX) to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected.

Interrupt Generation

In CAPT mode, an interrupt is generated on each filtered Fault A and Fault B and each dedicated CCy channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

Figure 40-30. Capture Action "CAPTMAX"

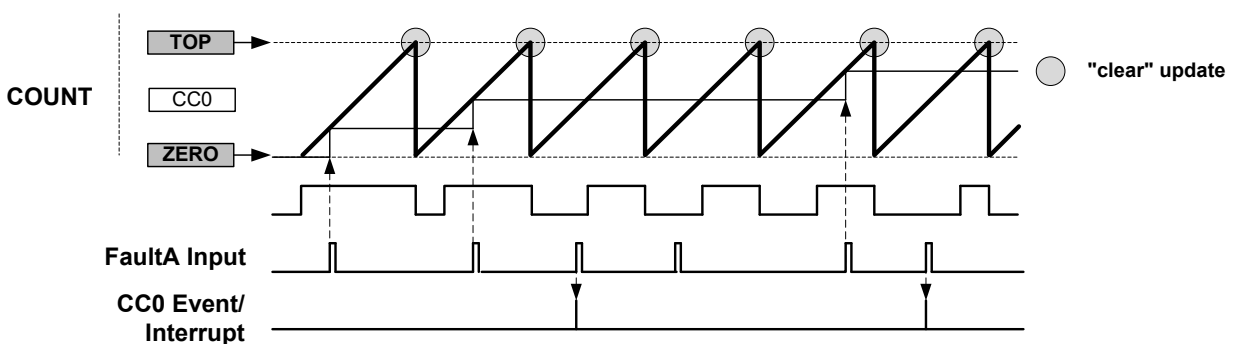
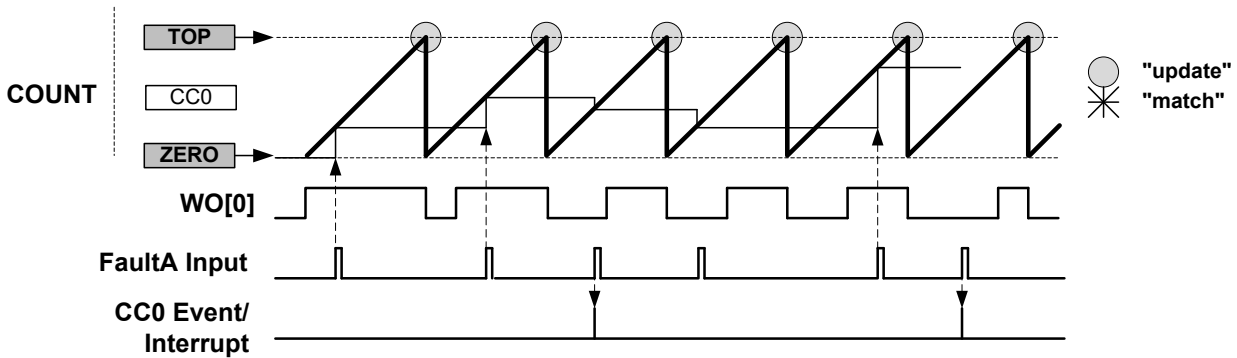


Figure 40-31. Capture Action “DERIVO”



Hardware Halt Action This is configured by writing 0x1 to the Fault A or Fault B Halt mode bits in the Recoverable Fault A or Fault B Configuration registers (FCTRLA.HALT(FCTRLA <9:8>) or FCTRLB.HALT(FCTRLB <9:8>). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The next figure ('Waveform Generation with Halt and Restart Actions') shows an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.

Figure 40-32. Waveform Generation with Halt and Restart Actions

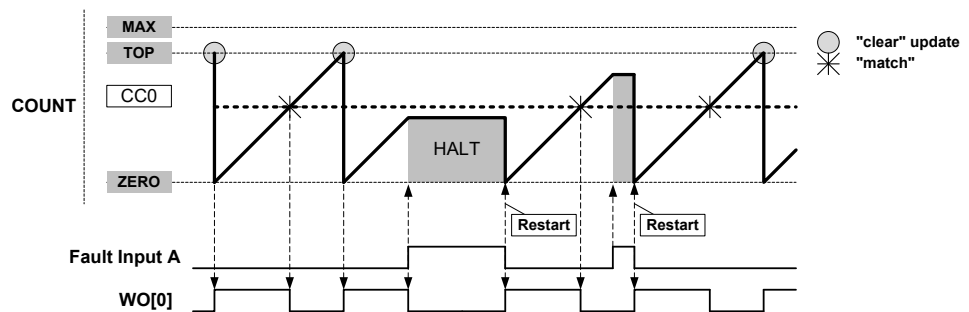
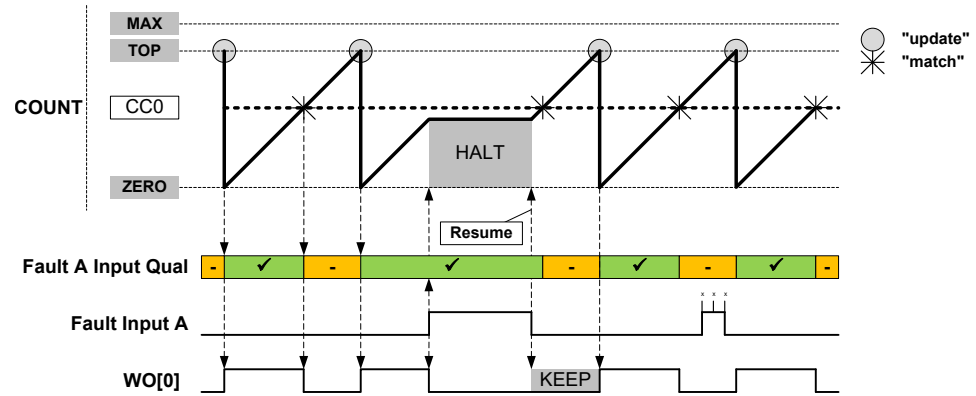
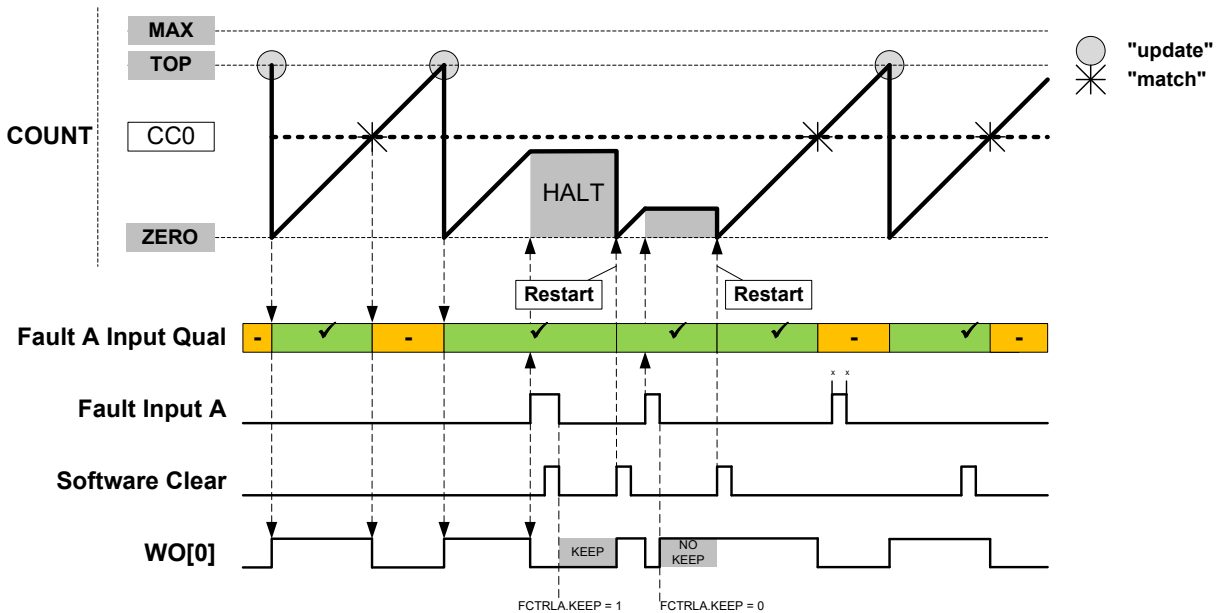


Figure 40-33. Waveform Generation with Fault Qualification, Halt, and Restart Actions



Software Halt Action This action is configured by writing 0x2 to the Fault A and Fault B Halt mode bits in the Recoverable Fault A or Fault B configuration registers (FCTRLA.HALT(FCTRLA <9:8>) or FCTRLB.HALT(FCTRLB <9:8>). Software halt action is similar to hardware halt action, but in order to restart the timer/counter, the corresponding fault condition must not be present anymore, and the corresponding FAULT A or Fault B bit in the STATUS register must be cleared by software. See the following figure.

Figure 40-34. Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions



40.5.3.6 Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NREy and DRVCTRL.NRVy). The non-recoverable fault input (TCCx_EV_0 and TCCx_EV_1) actions are enabled in Event Control register (EVCTRL.EVACT0(EVCTRL <2:0>) and EVCTRL.EVACT1(EVCTRL <5:3>)).

To avoid false fault detection on external events (e.g. a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input Filter Value bits in the Driver Control register DRVCTRL.FILTERVAL0 (DRVCTRL<27:24>) and DRVCTRL.FILTERVAL1 (DRVCTRL<31:28>). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD (DGBCTRL <2>)) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A, or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD (CTRLBSET <1>)=1 and the Ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

40.5.3.7 Time-Stamp Capture

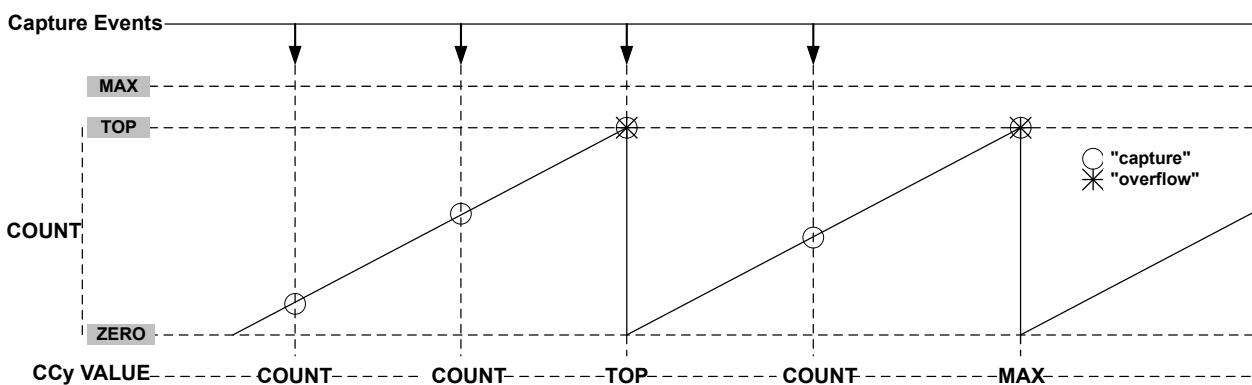
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT0(EVCTRL <2:0>)=0x6) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel y Compare/Capture Value (CCy) register. In case of an overflow, the MAX value is copied into the corresponding CCy register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel y Interrupt Flag (INTFLAG.MCy) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCy) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

Figure 40-35. Time-Stamp



40.5.3.8 Waveform Extension

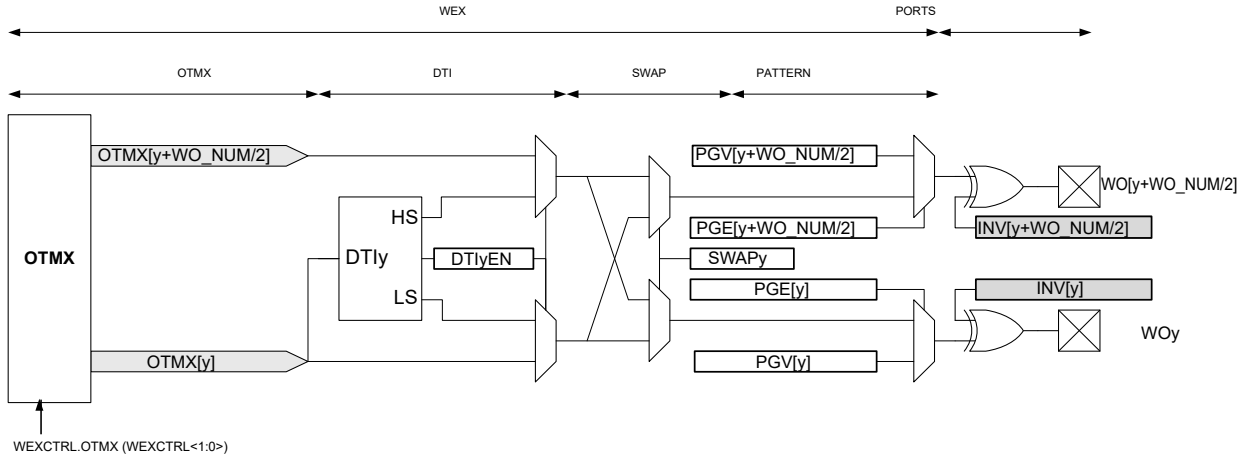
Waveform Extension Stage Details shows a schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

- Slice 0 DTI0 / SWAP0 acting on port pins (WO[0], WO[WO_NUM/2 +0])
- Slice 1 DTI1 / SWAP1 acting on port pins (WO[1], WO[WO_NUM/2 +1])

And generally:

- Slice y DTIy / SWAPy acting on port pins (WO[y], WO[WO_NUM/2 +y])

Figure 40-36. Waveform Extension Stage Details



The **output matrix (OTMX)** unit distributes compare channels, according to the selectable configurations in the following table. This is configured by WEXCTRL.OTMX bits (WEXCTRL <1:0>) as shown below.

Table 40-4. Output Matrix Channel Pin Routing Configuration

Value	OTMX[7]	OTMX[6]	OTMX[5]	OTMX[4]	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC1	CC0	CC5	CC4	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0	CC0	CC0	CC0	CC0
0x3	CC1	CC1	CC1	CC1	CC1	CC1	CC1	CC0

The following notes refer to this pin routing configuration:

- Configuration 0x0 is the default configuration. The channel location is the default one and channels are distributed on outputs modulo the number of channels. Channel 0 is routed to the Output matrix output OTMX[0], and Channel 1 to OTMX[1]. If there are more outputs than channels, then channel 0 is duplicated to the Output matrix output OTMX[CC_NUM], channel 1 to OTMX[CC_NUM+1] and so on.
- Configuration 0x1 distributes the channels on output modulo half the number of channels. This assigns twice the number of output locations to the lower channels than the default configuration. This can be used, for example, to control the four transistors of a full bridge using only two compare channels.
Using pattern generation, some of these four outputs can be overwritten by a constant level, enabling flexible drive of a full bridge in all quadrant configurations.
- Configuration 0x2 distributes compare channel 0 (CC0) to all port pins. With pattern generation, this configuration can control a stepper motor.
- Configuration 0x3 distributes the compare channel CC0 to the first output, and the channel CC1 to all other outputs. Together with pattern generation and the fault extension, this configuration can control up to seven LED strings, with a boost stage.

The table below is an example showing four compare channels on four outputs.

Table 40-5. Four Compare Channels on Four Outputs

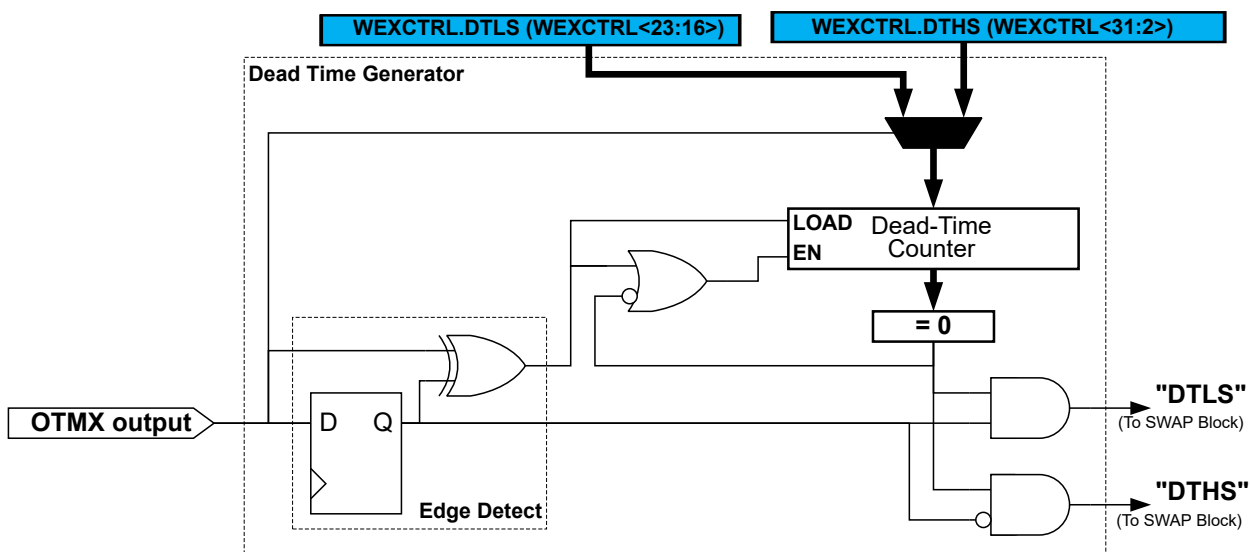
Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x0	CC3	CC2	CC1	CC0
0x1	CC1	CC0	CC1	CC0
0x2	CC0	CC0	CC0	CC0

Table 40-5. Four Compare Channels on Four Outputs (continued)

Value	OTMX[3]	OTMX[2]	OTMX[1]	OTMX[0]
0x3	CC1	CC1	CC1	CC0

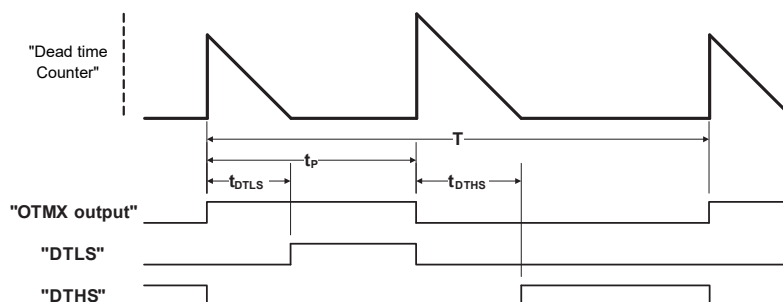
The **dead-time insertion (DTI)** unit generates OFF time with the non-inverted low side (LS) and inverted high side (HS) of the wave generator output forced at low level. This OFF time is called dead time. Dead-time insertion ensures that the LS and HS outputs (DTLS and DTHS) will never switch simultaneously.

The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. *Dead-Time Generator Block Diagram* shows the block diagram of one DTI generator. The four channels have a common register which controls the dead time, which is independent of high side and low side setting.

Figure 40-37. Dead-Time Generator Block Diagram

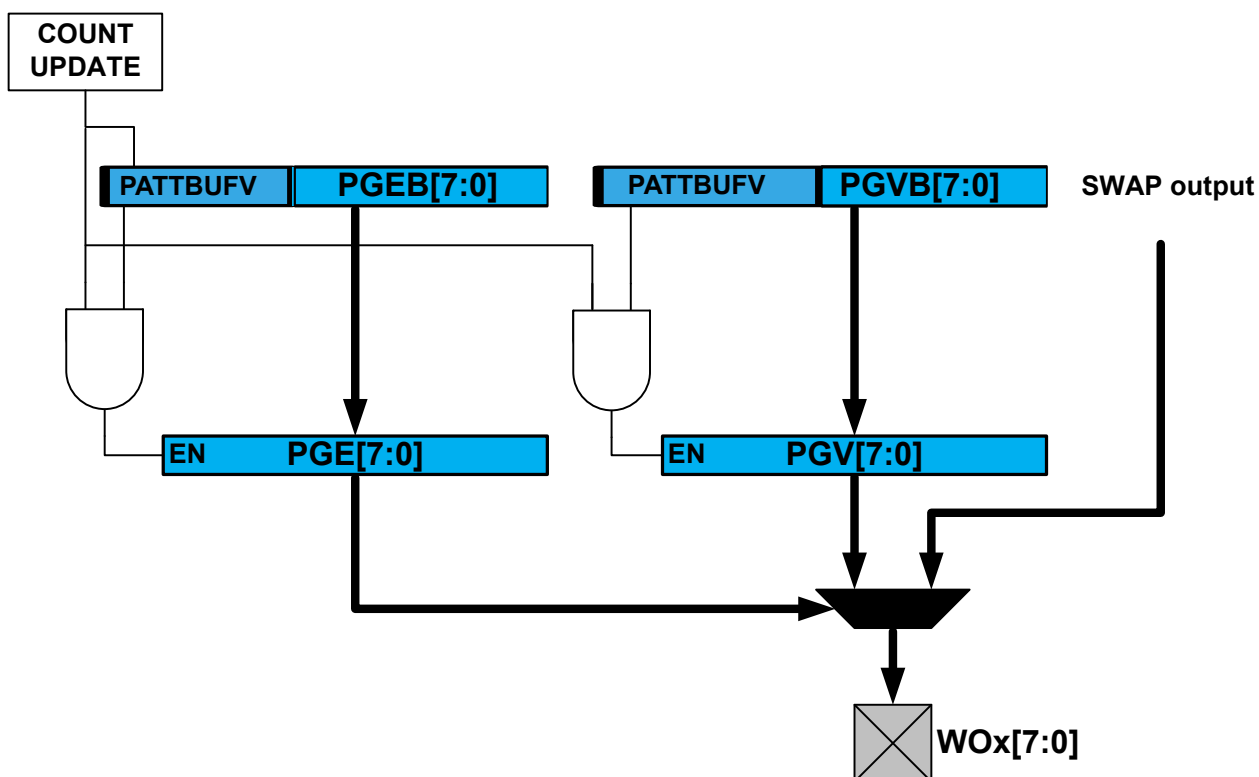
As shown in *Dead-Time Generator Timing Diagram*, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the output matrix (OTMX) output changes, the dead-time counter is reloaded according to the edge of the input. When the output changes from low to high (positive edge) it initiates a counter reload of the DTLS register. When the output changes from high to low (negative edge) it reloads the DTHS register.

In the following figure t_p shows the lower side output from OTMX and T is the period of OTMX output waveform. Dead-time insertion for lower and upper side outputs are shown by t_{DTLS} and t_{DTHS} respectively.

Figure 40-38. Dead-Time Generator Timing Diagram

The **pattern generator unit** produces a synchronized bit pattern across the port pins it is connected to. The pattern generation features are primarily intended for handling the commutation sequence in brushless DC motors (BLDC), stepper motors, and full bridge control. See also *Pattern Generator Block Diagram*.

Figure 40-39. Pattern Generator Block Diagram



As with other double-buffered timer/counter registers, the register update is synchronized to the UPDATE condition set by the timer/counter waveform generation operation. If synchronization is not required by the application, the software can simply access directly the **PGV[7:0]** bits (PATT <15:8>), **PGE[7:0]** bits (PATT <7:0>) registers.

40.5.4 DMA, Interrupts, and Events

The following table shows the DMA request for the TCC module.

Table 40-6. Module Requests for TCC

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
Overflow / Underflow	Yes	Yes		Yes ⁽¹⁾	On DMA acknowledge
Channel Compare Match or Capture	Yes	Yes	Yes ⁽²⁾	Yes ⁽³⁾	For circular buffering: on DMA acknowledge For capture channel: when CCy register is read
Retrigger	Yes	Yes			
Count	Yes	Yes			
Capture Overflow Error	Yes				
Debug Fault State	Yes				
Recoverable Faults	Yes				
Non-Recoverable Faults	Yes				

Table 40-6. Module Requests for TCC (continued)

Condition	Interrupt request	Event output	Event input	DMA request	DMA request is cleared
TCCx Event 0 input			Yes ⁽⁴⁾		
TCCx Event 1 input			Yes ⁽⁵⁾		

Notes:

1. DMA request set on Overflow, Underflow or Re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In Capture or Circular modes.
4. On event input, either action can be executed:
 - re-trigger counter
 - control counter direction
 - stop the counter
 - decrement the counter
 - perform period and pulse width capture
 - generate non-recoverable fault
5. On event input, either action can be executed:
 - re-trigger counter
 - increment or decrement counter depending on direction
 - start the counter
 - increment or decrement counter based on direction
 - increment counter regardless of direction
 - generate non-recoverable fault

40.5.4.1 DMA Operation

The TCC can generate the following DMA requests:

Counter overflow (OVF)	If the One-shot Trigger mode in the control A register (CTRLA.DMAOS(CTRLA<23>)) is written to '0', the TCC generates a DMA request on each cycle when an update condition (Overflow, Underflow or Re-trigger) is detected. When an update condition (Overflow, Underflow or Re-trigger) is detected while CTRLA.DMAOS(CTRLA<23>)=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD(CTRLBSET<7:5>)=DMAOS). In both cases, the request is cleared by hardware on DMA acknowledge.
Channel Match (MCy)	A DMA request is set only on a compare match if CTRLA.DMAOS(CTRLA<23>)=0. The request is cleared by hardware on DMA acknowledge. When CTRLA.DMAOS(CTRLA<23>)=1, the DMA requests are not generated.
Channel Capture (MCy)	For a capture channel, the request is set when valid data is present in the CCy register, and cleared once the CCy register is read. In this operation mode, the CTRLA.DMAOS bit (CTRLA<23>) value is ignored.



Important: DMA One-Shot mode is not available in RAMP1/RAMP2C/RAMP2CS modes.

DMA Operation with Circular Buffer

When circular buffer operation is enabled, the Buffer registers must be written in a correct order and synchronized to the update times of the timer. The DMA triggers of the TCC provide a way to ensure a safe and correct update of circular buffers.

Note: Circular buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

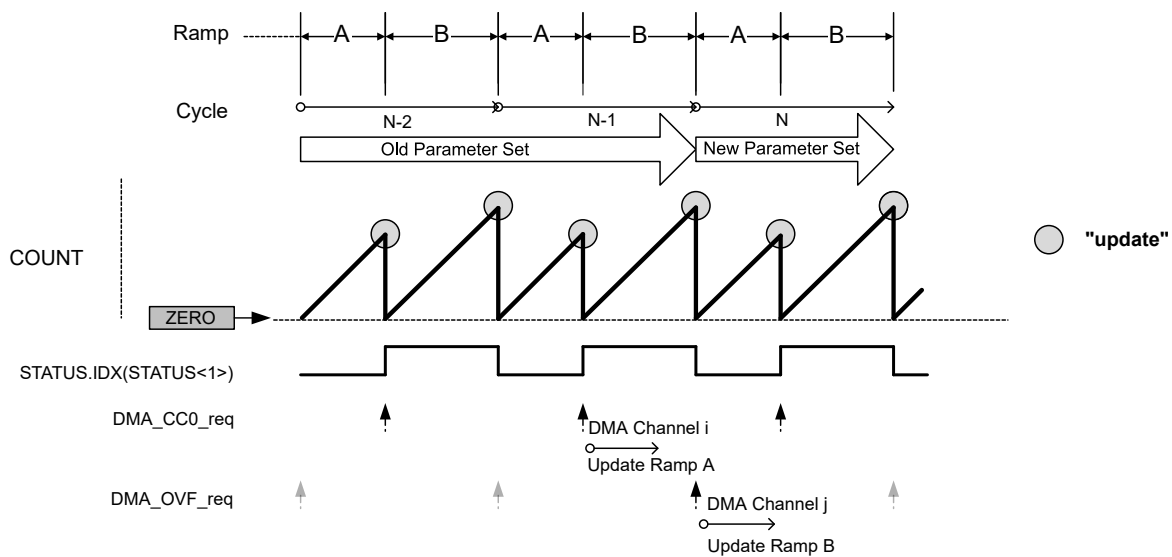
DMA Operation with Circular Buffer in RAMP2 and RAMP2A Operation

When a CC0 channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of Ramp B.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of Ramp A with an effective DMA transfer on previous Ramp B (DMA acknowledge).

The update of all circular buffer values for Ramp A can be done through a DMA channel triggered on a MC (Compare Match) trigger. The update of all circular buffer values for Ramp B, can be done through a second DMA channel triggered by the overflow DMA request.

Figure 40-40. DMA Triggers in RAMP2 Operation and Circular Buffer Enabled

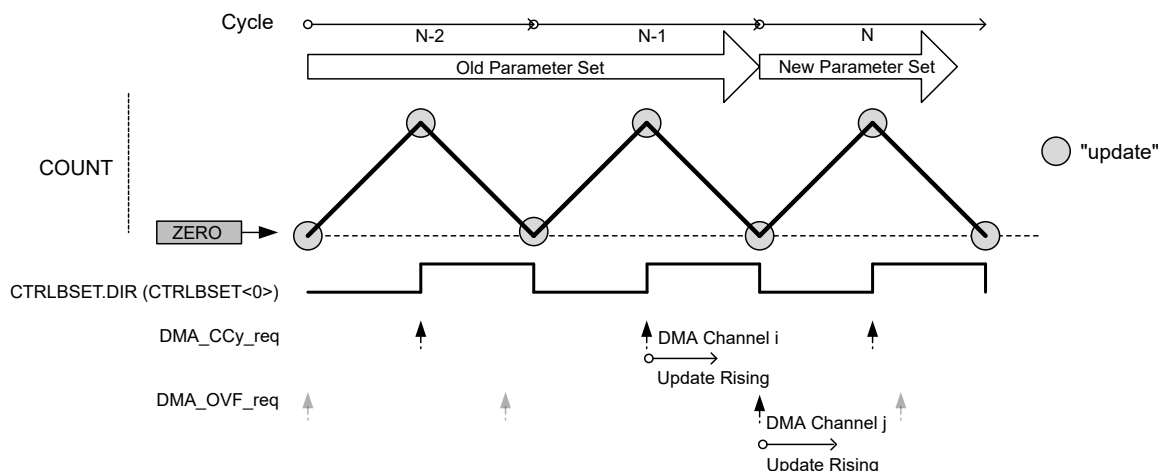


DMA Operation with Circular Buffer in DSBOTH

When CC0 channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of down-counting phase.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of up-counting phase with an effective DMA transfer on previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by MC (Compare Match) trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.

Figure 40-41. DMA Triggers in DSBOOTH Operation and Circular Buffer Enabled

40.5.4.2 Interrupts

The TCC has the following interrupt sources (refer to the INTENSET register):

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Non-Recoverable Update Fault (UFS)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTA and FAULTB)
- Non-recoverable Faults (FAULT0 and FAULT1)
- Compare Match or Capture Channels (MCy)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/ Sleep Mode Controller section for details.

References:

- PM_Sleep Mode Controller

Each interrupt source has an Interrupt flag associated with it. The Interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the Interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled, or the TCC is reset. Refer to the INTFLAG for details on how to clear Interrupt flags. The TCC has one common interrupt request line for OVF, ERR, UFS, DFS, FAULTA, FAULTB, FAULT0, and FAULT1 interrupt sources. It contains a separate line for both TRG, and CNT interrupt sources and another line for all Compare Match or Capture Channels (MCy) interrupt sources. The user must read the INTFLAG register to determine which Interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

References:

- Nested Vector Interrupt Controller

40.5.4.3 Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) For further details, refer to EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCy

Writing a '1' or '0' to an Event Output bit in the Event Control Register (OVFEO, TRGEO, CNTEO and MCEOy bits) enables or disables the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCy):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx_EV_1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx_EV_0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start - start counting on the event rising edge. Further events will not restart the counter; the counter will keep on counting using prescaled GCLK_TCCx, until it reaches TOP or ZERO, depending on the direction.
- Counter increment on event. This will increment the counter, irrespective of the counter direction.
- Count during active state of an asynchronous event (increment or decrement, depending on counter direction). In this case, the counter will be incremented or decremented on each cycle of the prescaled clock, as long as the event is active.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0(EVCTRL <2:0>) and EVCTRL.EVACT1(EVCTRL <5:3>)). For further details, refer to EVCTRL.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.MCEly or EVCTRL.TCEIn where n=0,1) enables the corresponding action on input event, where writing '0' disables the action on input event.

Note: When several events are connected to the TCC, the enabled action will apply for each of the incoming events. Refer to *EVSYS – Event System* for details on how to configure the event system.

Reference:

- EVSYS

40.5.5 Sleep Mode Operation

The TCC can be configured to operate in any Sleep mode. To be able to run in standby the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. The TCC interrupt can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

References:

- EVSYS

40.6 Peripheral Dependencies

Table 40-7. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
TCC0	0x44810000	88 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 89 : CNT/TRG 90 : MC0 91 : MC1	APB: CLKMSK[1] MASK8	PCHCTRL[25]	30 STATUS0 PERID30	12 : OVF 13 : MC0 14 : MC1	<u>Generators:</u> OVF: CHANNEL[41] TRG: CHANNEL[42] CNT: CHANNEL[43] MC0: CHANNEL[44] MC1: CHANNEL[45] <u>Users:</u> EV0: USER[10],Resynch EV1: USER[11],Resynch MC0: USER[12],Resynch MC1: USER[13],Resynch	PD_CORE_SW
TCC1	0x44812000	92 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 93 : CNT/TRG 94 : MC0 95 : MC1	APB: CLKMSK[1] MASK9	PCHCTRL[25]	31 STATUS0 PERID31	15 : OVF 16 : MC0 17 : MC1	<u>Generators:</u> OVF: CHANNEL[46] TRG: CHANNEL[47] CNT: CHANNEL[48] MC0: CHANNEL[49] MC1: CHANNEL[50] <u>Users:</u> EV0: USER[14],Resynch EV1: USER[15],Resynch MC0: USER[16],Resynch MC1: USER[17],Resynch	PD_CORE_SW

Table 40-7. Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
TCC2	0x44814000	96 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 97 : CNT/TRG 98 : MC0 99 : MC1	APB: CLKMSK[1] MASK10	PCHCTRL[26]	32 STATUS1 PERID0	18 : OVF 19 : MC0 20 : MC1	<u>Generators:</u> OVF: CHANNEL[51] TRG: CHANNEL[52] CNT: CHANNEL[53] MC0: CHANNEL[54] MC1: CHANNEL[55] <u>Users:</u> EV0: USER[18],Resynch EV1: USER[19],Resynch MC0: USER[20],Resynch MC1: USER[21],Resynch	PD_CORE_SW
TCC3	0x44816000	100 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 101 : CNT/TRG 102 : MC0 103 : MC1	APB: CLKMSK[1] MASK11	PCHCTRL[27]	33 STATUS1 PERID1	21 : OVF 22 : MC0 23 : MC1	<u>Generators:</u> OVF: CHANNEL[56] TRG: CHANNEL[57] CNT: CHANNEL[58] MC0: CHANNEL[59] MC1: CHANNEL[60] <u>Users:</u> EV0: USER[22],Resynch EV1: USER[23],Resynch MC0: USER[24],Resynch MC1: USER[25],Resynch	PD_CORE_SW

Table 40-7. Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
TCC4	0x44824000	118 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 119 : CNT/TRG 120 : MC0 121 : MC1	APB: CLKMSK[1] MASK14	PCHCTRL[30]	36 STATUS1 PERID4	28 : OVF 29 : MC0 30 : MC1	<u>Generators:</u> OVF: CHANNEL[61] TRG: CHANNEL[62] CNT: CHANNEL[63] MC0: CHANNEL[64] MC1: CHANNEL[65] <u>Users:</u> EV0: USER[26],Resynch EV1: USER[27],Resynch MC0: USER[28],Resynch MC1: USER[29],Resynch	PD_CORE_SW
TCC5	0x44826000	122 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 123 : CNT/TRG 124 : MC0 125 : MC1	APB: CLKMSK[1] MASK15	PCHCTRL[31]	37 STATUS1 PERID5	31 : OVF 32 : MC0 33 : MC1	<u>Generators:</u> OVF: CHANNEL[66] TRG: CHANNEL[67] CNT: CHANNEL[68] MC0: CHANNEL[69] MC1: CHANNEL[70] <u>Users:</u> EV0: USER[30],Resynch EV1: USER[31],Resynch MC0: USER[32],Resynch MC1: USER[33],Resynch	PD_CORE_SW

Table 40-7. Peripheral Dependencies (continued)

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
TCC6	0x44828000	126 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS 127 : CNT/TRG 128 : MC0 129 : MC1	APB: CLKMSK[1] MASK16	PCHCTRL[32]	38 STATUS1 PERID6	34 : OVF 35 : MC0 36 : MC1	<u>Generators:</u> OVF: CHANNEL[71] TRG: CHANNEL[72] CNT: CHANNEL[73] MC0: CHANNEL[74] MC1: CHANNEL[75] <u>Users:</u> EV0: USER[34],Resynch EV1: USER[35],Resynch MC0: USER[36],Resynch MC1: USER[37],Resynch	PD_CORE_SW

40.7 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24	CPTEN[7:0]							
		23:16	DMAOS							FCYCLE
		15:8	MSYNC	ALOCK	PRESCSYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
		7:0		RESOLUTION[1:0]				ENABLE	SWRST	
0x04	CTRLBCLR	7:0	CMD[2:0]		IDXCMD[1:0]		ONESHOT	LUPD	DIR	
0x05	CTRLBSET	7:0	CMD[2:0]		IDXCMD[1:0]		ONESHOT	LUPD	DIR	
0x06	Reserved									
... 0x07										
0x08	SYNCBUSY	31:24								
		23:16								
		15:8	CC[7:0]							
		7:0	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
0x0C	FCTRLA	31:24					FILTERVAL[3:0]			
		23:16	BLANKVAL[7:0]							
		15:8	BLANKPRESC	CAPTURE[2:0]		CHSEL[1:0]		HALT[1:0]		
		7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x10	FCTRLB	31:24					FILTERVAL[3:0]			
		23:16	BLANKVAL[7:0]							
		15:8	BLANKPRESC	CAPTURE[2:0]		CHSEL[1:0]		HALT[1:0]		
		7:0	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
0x14	WEXCTRL	31:24	DTHS[7:0]							
		23:16	DTLS[7:0]							
		15:8					DTIEN[3:0]			
		7:0						OTMX[1:0]		
0x18	DRVCTRL	31:24	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
		23:16	INVEN[7:0]							
		15:8	NRV[7:0]							
		7:0	NRE[7:0]							
0x1C ... 0x1D	Reserved									
0x1E	DBGCTRL	7:0						FDDBD		DBGRUN
0x1F	Reserved									
0x20	EVCTRL	31:24	MCEO[7:0]							
		23:16	MCEI[7:0]							
		15:8	TCEI[1:0]		TCINV[1:0]			CNTEO	TRGEO	OVFEO
		7:0	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
0x24	INTENCLR	31:24								
		23:16	MC[7:0]							
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x28	INTENSET	31:24								
		23:16	MC[7:0]							
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x2C	INTFLAG	31:24								
		23:16	MC[7:0]							
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
		7:0					ERR	CNT	TRG	OVF
0x30	STATUS	31:24	CMP[7:0]							
		23:16	CCBUFV[7:0]							
		15:8	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
		7:0	PERBUFV		PATTBUFV	SLAVE	DFS	UFS	IDX	STOP
0x34	COUNT	31:24	COUNT[31:24]							
		23:16	COUNT[23:16]							
		15:8	COUNT[15:8]							
		7:0	COUNT[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x38	PATT	15:8	PGV[7:0]							
		7:0	PGE[7:0]							
0x3A ... 0x3B	Reserved									
0x3C	WAVE	31:24	SWAP[3:0]							
		23:16	POL[7:0]							
		15:8	CICCEN[3:0]							
		7:0	CIPEREN	RAMP[2:0]				WAVEGEN[2:0]		
0x40	PER	31:24	PER[31:24]							
		23:16	PER[23:16]							
		15:8	PER[15:8]							
		7:0	PER[7:0]							
0x44	CC0	31:24	CC[31:24]							
		23:16	CC[23:16]							
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x48	CC1	31:24	CC[31:24]							
		23:16	CC[23:16]							
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x4C	CC2	31:24	CC[31:24]							
		23:16	CC[23:16]							
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x50	CC3	31:24	CC[31:24]							
		23:16	CC[23:16]							
		15:8	CC[15:8]							
		7:0	CC[7:0]							
0x54 ... 0x63	Reserved									
0x64	PATTBUF	15:8	PGVB[7:0]							
		7:0	PGE[7:0]							
0x66 ... 0x6B	Reserved									
0x6C	PERBUF	31:24	PERBUF[31:24]							
		23:16	PERBUF[23:16]							
		15:8	PERBUF[15:8]							
		7:0	PERBUF[7:0]							
0x70	CCBUF0	31:24	CCBUF[31:24]							
		23:16	CCBUF[23:16]							
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							
0x74	CCBUF1	31:24	CCBUF[31:24]							
		23:16	CCBUF[23:16]							
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							
0x78	CCBUF2	31:24	CCBUF[31:24]							
		23:16	CCBUF[23:16]							
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							
0x7C	CCBUF3	31:24	CCBUF[31:24]							
		23:16	CCBUF[23:16]							
		15:8	CCBUF[15:8]							
		7:0	CCBUF[7:0]							

40.7.1 Control A

Name: CTRLA
Offset: 0x00
Reset: 0x00000000
Property: RW

Table 40-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CPTEN[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DMAOS							FCYCLE
Access								
Reset	0							0
Bit	15	14	13	12	11	10	9	8
	MSYNC	ALOCK	PRESCSYNC[1:0]		RUNSTDBY	PRESCALER[2:0]		
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
		RESOLUTION[1:0]					ENABLE	SWRST
Access								
Reset		0	0				0	0

Bits 31:24 – CPTEN[7:0] Capture Channel x Enable

Bit 23 – DMAOS DMA One-shot Trigger Mode

Bit 16 – FCYCLE Full Cycle

Bit 15 – MSYNC Master Synchronization (only for TCC Slave Instance)

Bit 14 – ALOCK Auto Lock

Bits 13:12 – PRESCSYNC[1:0] Prescaler and Counter Synchronization Selection

Value	Name	Description
0	GCLK	Reload or reset counter on next GCLK
1	PRESC	Reload or reset counter on next prescaler clock
2	RESYNC	Reload or reset counter on next GCLK and reset prescaler counter

Bit 11 – RUNSTDBY Run in Standby

Bits 10:8 – PRESCALER[2:0] Prescaler

Value	Name	Description
0	DIV1	No division
1	DIV2	Divide by 2
2	DIV4	Divide by 4
3	DIV8	Divide by 8
4	DIV16	Divide by 16
5	DIV64	Divide by 64
6	DIV256	Divide by 256
7	DIV1024	Divide by 1024

Bits 6:5 – RESOLUTION[1:0] Enhanced Resolution

Value	Name	Description
0	NONE	Dithering is disabled
1	DITH4	Dithering is done every 16 PWM frames
2	DITH5	Dithering is done every 32 PWM frames
3	DITH6	Dithering is done every 64 PWM frames

Bit 1 – ENABLE Enable

Bit 0 – SWRST Software Reset

40.7.2 Control B Clear

Name: CTRLBCLR
Offset: 0x04
Reset: 0x00
Property: RW

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBSET) register.

Table 40-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access								
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0] TCC Command

These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will read back zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.

Writing zero to this bit group has no effect.

Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0	NONE	No action
1	RETRIGGER	Clear start, restart or retrigger
2	STOP	Force stop
3	UPDATE	Force update or double buffered registers
4	READSYNC	Force COUNT read synchronization
5	DMAOS	One-shot DMA trigger

Bits 4:3 – IDXCMD[1:0] Ramp Index Command

These bits can be used to force cycle A (Ramp A) and cycle B (Ramp B) changes in all RAMP2x operations. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.

Writing zero to these bits has no effect.

Writing a '1' to any of these bits will clear the pending command.

Value	Name	Description
0	DISABLE	Command disabled: Index toggles between cycles A and B
1	SET	Set index: cycle B will be forced in the next cycle
2	CLEAR	Clear index: cycle A will be forced in the next cycle
3	HOLD	Hold index: the next cycle will be the same as the current cycle

Bit 2 – ONESHOT One-Shot

This bit controls one-shot operation of the TCC. When one-shot operation is enabled, the TCC will stop counting on the next overflow/underflow condition or on a stop command.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will disable the one-shot operation.

Reading this bit gives the following status:

Value	Description
0	The TCC will update the counter value on overflow/underflow condition and continue operation.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update

This bit controls the update operation of the TCC buffered registers.

When CTRLBCLR.LUPD (CTRLBCLR<1>) is cleared, the hardware UPDATE registers with value from their buffered registers is enabled.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable the registers updates on hardware UPDATE condition.

Reading this bit gives the following status:

Value	Description
0	The CCBUFy, PERBUF buffer register bitfields <i>are</i> copied into the corresponding CCy, PER registers and register bitfields on hardware update condition.
1	The CCBUFy, PERBUF buffer registers bitfields are <i>not</i> copied into the corresponding CCy, PER registers and register bitfields on hardware update condition.

Bit 0 – DIR Counter Direction

This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will clear the bit and make the counter count up. Reading this bit gives the following status:

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.7.3 Control B Set

Name: CTRLBSET
Offset: 0x05
Reset: 0x00
Property: RW

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBCLR) register.

Table 40-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
	CMD[2:0]			IDXCMD[1:0]		ONESHOT	LUPD	DIR
Access								
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – CMD[2:0] TCC Command

These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.

Writing zero to this bit group has no effect

Writing a valid value to this bit group will set the associated command.

Value	Name	Description
0	NONE	No action
1	RETRIGGER	Clear start, restart or retrigger
2	STOP	Force stop
3	UPDATE	Force update or double buffered registers
4	READSYNC	Force COUNT read synchronization
5	DMAOS	One-shot DMA trigger

Bits 4:3 – IDXCMD[1:0] Ramp Index Command

These bits can be used to force cycle A and cycle B changes in all RAMP2x operations. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.

Writing a zero to these bits has no effect.

Writing a valid value to these bits will set a command.

Value	Name	Description
0	DISABLE	Command disabled: Index toggles between cycles A and B
1	SET	Set index: cycle B will be forced in the next cycle
2	CLEAR	Clear index: cycle A will be forced in the next cycle
3	HOLD	Hold index: the next cycle will be the same as the current cycle

Bit 2 – ONESHOT One-Shot

This bit controls one-shot operation of the TCC. When in one-shot operation, the TCC will stop counting on the next overflow/underflow condition or a stop command.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will enable the one-shot operation.

Value	Description
0	The TCC will count continuously.
1	The TCC will stop counting on the next underflow/overflow condition.

Bit 1 – LUPD Lock Update

This bit controls the update operation of the TCC buffered registers.

When CTRLBSET.LUPD (CTRLBSET<1>) is set, the hardware UPDATE registers with value from their buffered registers is disabled. Disabling the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.

This bit has no effect when input capture operation is enabled.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will disable the registers updates on hardware UPDATE condition.

Value	Description
0	The CCBUFy, PERBUF buffer register bitfields <i>are</i> copied into the corresponding CCy, PER registers and register bitfields on hardware update condition.
1	The CCBUFy, PERBUF buffer registers bitfields are <i>not</i> copied into the corresponding CCy, PER registers and register bitfields on hardware update condition.

Bit 0 – DIR Counter Direction

This bit is used to change the direction of the counter.

Writing a '0' to this bit has no effect

Writing a '1' to this bit will set the bit and make the counter count down. Reading this bit gives following status:

Value	Description
0	The timer/counter is counting up (incrementing).
1	The timer/counter is counting down (decrementing).

40.7.4 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: R

Table 40-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	CC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PER	WAVE	PATT	COUNT	STATUS	CTRLB	ENABLE	SWRST
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – CC[7:0] Compare Channel x Busy

Bit 7 – PER Period Busy

This bit is cleared when the synchronization of PER register between the clock domains is complete.
This bit is set when the synchronization of PER register between clock domains is started.

Bit 6 – WAVE Wave Busy

This bit is cleared when the synchronization of WAVE register between the clock domains is complete.
This bit is set when the synchronization of WAVE register between clock domains is started.

Bit 5 – PATT Pattern Busy

Bit 4 – COUNT Count Busy

This bit is cleared when the synchronization of COUNT register between the clock domains is complete.
This bit is set when the synchronization of COUNT register between clock domains is started.

Bit 3 – STATUS Status Busy

This bit is cleared when the synchronization of STATUS register between the clock domains is complete.

This bit is set when the synchronization of STATUS register between clock domains is started.

Bit 2 – CTRLB Ctrlb Busy

This bit is cleared when the synchronization of CTRLBSET/CTRLBCLR register between the clock domains is complete.

This bit is set when the synchronization of CTRLBSET/CTRLBCLR register between clock domains is started.

Bit 1 – ENABLE Enable Busy

This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.

This bit is set when the synchronization of ENABLE bit between clock domains is started.

Bit 0 – SWRST Swrst Busy

This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.

This bit is set when the synchronization of SWRST bit between clock domains is started.

40.7.5 Recoverable Fault A Configuration

Name: FCTRLA
Offset: 0x0C
Reset: 0x00000000
Property: RW

Table 40-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						FILTERVAL[3:0]		
Access								
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BLANKVAL[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access								
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0] Fault A Filter Value

These bits define the filter value applied on the Fault A (MCE0) event input line. Input signal with pulse width shorter than $(\text{FILTERVAL}) * \text{GCLK\_TCCx}$ is filtered. An Input signal with pulse width larger than $(\text{FILTERVAL}+1) * \text{GCLK\_TCCx}$ is passed to the TCC. The value must be set to zero when this event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Fault A Blanking Time

These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLA.BLANK(FCTRLA<6:5>)). When enabled, the fault input source is internally disabled for $\text{BLANKVAL} * \text{prescaled GCLK\_TCCx}$ periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Fault A Blanking Prescaler

This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

Value	Description
0	Blank time is $\text{BLANKVAL} * \text{prescaled GCLK\_TCCx}$.
1	Blank time is $\text{BLANKVAL} * 64 * \text{prescaled GCLK\_TCCx}$.

Bits 14:12 – CAPTURE[2:0] Fault A Capture Action

These bits select the capture and Fault A interrupt/event conditions.

Table 40-13. Fault A Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault A is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) , if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local minimum detection.
0x3	CAPTMAX	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) , if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local minimum value detection.
0x5	LOCMAX	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum detection.
0x6	DERIVO	On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum or minimum detection.
0x7	CAPTMARK	Capture with Ramp index as MSB value.

Value	Name	Description
0	DISABLE	No capture
1	CAPT	Capture on fault
2	CAPTMIN	Minimum capture
3	CAPTMAX	Maximum capture
4	LOCMIN	Minimum local detection
5	LOCMAX	Maximum local detection
6	DERIVO	Minimum and maximum local detection
7	CAPTMARK	Capture with ramp index as MSB value

Bits 11:10 – CHSEL[1:0] Fault A Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault A.

Value	Name	Description
0	CC0	Capture value stored in channel 0
1	CC1	Capture value stored in channel 1
2	CC2	Capture value stored in channel 2
3	CC3	Capture value stored in channel 3

Bits 9:8 – HALT[1:0] Fault A Halt Mode

These bits select the halt action for recoverable Fault A.

Value	Name	Description
0	DISABLE	Halt action disabled
1	HW	Hardware halt action
2	SW	Software halt action
3	NR	Non-recoverable fault

Bit 7 – RESTART Fault A Restart

Setting this bit enables restart action for Fault A.

Value	Description
0	Fault A restart action is disabled.
1	Fault A restart action is enabled.

Bits 6:5 – BLANK[1:0] Fault A Blanking Mode

These bits, select the blanking start point for recoverable Fault A.

Value	Name	Description
0	START	Blanking applied from start of the ramp
1	RISE	Blanking applied from rising edge of the output waveform
2	FALL	Blanking applied from falling edge of the output waveform
3	BOTH	Blanking applied from each toggle of the output waveform

Bit 4 – QUAL Fault A Qualification

Setting this bit enables the recoverable Fault A input qualification.

Value	Description
0	The recoverable Fault A input is not disabled on CMPy value condition.
1	The recoverable Fault A input is disabled when channel output is at inactive level (CMPy = 0).

Bit 3 – KEEP Fault A Keeper

Setting this bit enables the Fault A keep action.

Value	Description
0	The Fault A state is released as soon as the recoverable Fault A is released.
1	The Fault A state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0] Fault A Source

These bits select the TCC Fault A mode.

Event system channel connected to Match or Capture Channel 0 (MCE0) event input, must be configured to route the event asynchronously, when used as a recoverable Fault A input.

Value	Name	Description
0	DISABLE	Fault input disabled
1	ENABLE	MCEx (x=0,1) event input
2	INVERT	Inverted MCEx (x=0,1) event input
3	ALTFault	Alternate fault (A or B) state at the end of the previous period

40.7.6 Recoverable Fault B Configuration

Name: FCTRLB
Offset: 0x10
Reset: 0x00000000
Property: RW

Table 40-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						FILTERVAL[3:0]		
Access								
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BLANKVAL[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	BLANKPRESC	CAPTURE[2:0]			CHSEL[1:0]		HALT[1:0]	
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	RESTART	BLANK[1:0]		QUAL	KEEP		SRC[1:0]	
Access								
Reset	0	0	0	0	0		0	0

Bits 27:24 – FILTERVAL[3:0] Fault B Filter Value

These bits define the filter value applied on FAULT B (MCE1) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. The value must be set to zero when this event is used as synchronous event.

Bits 23:16 – BLANKVAL[7:0] Fault B Blanking Time

These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLB.BLANK(FCTRLB<6:5>)). When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCCx periods after the detection of the waveform edge.

Bit 15 – BLANKPRESC Fault B Blanking Prescaler

This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

Value	Description
0	Blank time is BLANKVAL* prescaled GCLK_TCCx.
1	Blank time is BLANKVAL* 64 * prescaled GCLK_TCCx.

Bits 14:12 – CAPTURE[2:0] Fault B Capture Action

These bits select the capture and Fault B interrupt/event conditions.

Table 40-15. Fault B Capture Action

Value	Name	Description
0x0	DISABLE	Capture on valid recoverable Fault B is disabled
0x1	CAPT	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each new captured value.
0x2	CAPTMIN	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>), if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local minimum detection.
0x3	CAPTMAX	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>), if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local maximum detection.
0x4	LOCMIN	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local minimum value detection.
0x5	LOCMAx	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local maximum detection.
0x6	DERIVO	On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local maximum or minimum detection.
0x7	CAPTMARK	Capture with Ramp index as MSB value.

Value	Name	Description
0	DISABLE	No capture
1	CAPT	Capture on fault
2	CAPTMIN	Minimum capture
3	CAPTMAX	Maximum capture
4	LOCMIN	Minimum local detection
5	LOCMAx	Maximum local detection
6	DERIVO	Minimum and maximum local detection
7	CAPTMARK	Capture with ramp index as MSB value

Bits 11:10 – CHSEL[1:0] Fault B Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault B.

Value	Name	Description
0	CC0	Capture value stored in channel 0
1	CC1	Capture value stored in channel 1
2	CC2	Capture value stored in channel 2
3	CC3	Capture value stored in channel 3

Bits 9:8 – HALT[1:0] Fault B Halt Mode

These bits select the halt action for recoverable Fault B.

Value	Name	Description
0	DISABLE	Halt action disabled
1	HW	Hardware halt action
2	SW	Software halt action
3	NR	Non-recoverable fault

Bit 7 – RESTART Fault B Restart

Setting this bit enables restart action for Fault B.

Value	Description
0	Fault B restart action is disabled.
1	Fault B restart action is enabled.

Bits 6:5 – BLANK[1:0] Fault B Blanking Mode

These bits, select the blanking start point for recoverable Fault B.

Value	Name	Description
0	START	Blanking applied from start of the ramp
1	RISE	Blanking applied from rising edge of the output waveform
2	FALL	Blanking applied from falling edge of the output waveform
3	BOTH	Blanking applied from each toggle of the output waveform

Bit 4 – QUAL Fault B Qualification

Setting this bit enables the recoverable Fault B input qualification.

Value	Description
0	The recoverable Fault B input is not disabled on CMPy value condition.
1	The recoverable Fault B input is disabled when output signal is at inactive level (CMPy = 0).

Bit 3 – KEEP Fault B Keeper

Setting this bit enables the Fault B keep action.

Value	Description
0	The Fault B state is released as soon as the recoverable Fault B is released.
1	The Fault B state is released at the end of TCC cycle.

Bits 1:0 – SRC[1:0] Fault B Source

These bits select the TCC Fault B mode.

Event system channel connected to Match or Capture Channel 1 (MCE1) event input, must be configured to route the event asynchronously, when used as a recoverable Fault B input.

Value	Name	Description
0	DISABLE	Fault input disabled
1	ENABLE	MCEx (x=0,1) event input
2	INVERT	Inverted MCEx (x=0,1) event input
3	ALTFault	Alternate fault (A or B) state at the end of the previous period

40.7.7 Waveform Extension Configuration

Name: WEXCTRL
Offset: 0x14
Reset: 0x00000000
Property: RW

Table 40-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DTHS[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DTLS[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
					DTIEN[3:0]			
Access								
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
							OTMX[1:0]	
Access								
Reset							0	0

Bits 31:24 – DTHS[7:0] Dead-time High Side Outputs Value

Bits 23:16 – DTLS[7:0] Dead-time Low Side Outputs Value

Bits 11:8 – DTIEN[3:0] Dead-time Insertion Generator x Enable

Bits 1:0 – OTMX[1:0] Output Matrix

40.7.8 Driver Control

Name: DRVCTRL
Offset: 0x18
Reset: 0x00000000
Property: RW

Table 40-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FILTERVAL1[3:0]				FILTERVAL0[3:0]			
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	INVEN[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	NRV[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NRE[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:28 – FILTERVAL1[3:0] Non-Recoverable Fault Input 1 Filter Value

These bits define the filter value applied on Fault 1 (TCE1) event input line. An Input signal with pulse width shorter than $(\text{FILTERVAL}) * \text{GCLK\_TCCx}$ is filtered. An Input signal with pulse width larger than $(\text{FILTERVAL}+1) * \text{GCLK\_TCCx}$ is passed. When value is 0, Fault1 acts asynchronously on PWM outputs.

Bits 27:24 – FILTERVAL0[3:0] Non-Recoverable Fault Input 0 Filter Value

These bits define the filter value applied on Fault 0 (TCE0) event input line. An Input signal with pulse width shorter than $(\text{FILTERVAL}) * \text{GCLK\_TCCx}$ is filtered. An Input signal with pulse width larger than $(\text{FILTERVAL}+1) * \text{GCLK\_TCCx}$ is passed. When value is 0, Fault 0 acts asynchronously on PWM outputs.

Bits 23:16 – INVEN[7:0] Output Waveform x Inversion

Bits 15:8 – NRV[7:0] Non-Recoverable State x Output Value

Bits 7:0 – NRE[7:0] Non-Recoverable State x Output Enable

40.7.9 Debug Control

Name: DBGCTRL
Offset: 0x1E
Reset: 0x00
Property: RW

Table 40-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	7	6	5	4	3	2	1	0
						FDDBD		DBGRUN
Access						0		0
Reset								

Bit 2 – FDDBD Fault Detection on Debug Break Detection

This bit is not affected by software Reset and should not be changed by software while the TCC is enabled.

By default this bit is zero, and the on-chip debug (OCD) fault protection is disabled. When this bit is written to '1', OCD break request from the OCD system will trigger non-recoverable fault.

Value	Description
0	No faults are generated when TCC is halted in Debug mode.
1	A non recoverable fault is generated and INTFLAG.DFS (INTFLAG<11>) flag is set when TCC is halted in Debug mode.

Bit 0 – DBGRUN Debug Running Mode

This bit is not affected by software Reset and should not be changed by software while the TCC is enabled.

Value	Description
0	The TCC is halted when the device is halted in Debug mode.
1	The TCC continues normal operation when the device is halted in Debug mode.

40.7.10 Event Control

Name: EVCTRL
Offset: 0x20
Reset: 0x00000000
Property: RW

Table 40-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MCEO[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MCEI[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TCEI[1:0]		TCINV[1:0]			CNTEO	TRGEO	OVFEO
Access								
Reset	0	0	0	0		0	0	0
Bit	7	6	5	4	3	2	1	0
	CNTSEL[1:0]		EVACT1[2:0]			EVACT0[2:0]		
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:24 – MCEO[7:0] Match or Capture Channel x Event Output Enable

Bits 23:16 – MCEI[7:0] Match or Capture Channel x Event Input Enable

Bits 15:14 – TCEI[1:0] Timer/counter Event x Input Enable

Bits 13:12 – TCINV[1:0] Inverted Event x Input Enable

Bit 10 – CNTEO Timer/counter Output Event Enable

This bit is used to enable the counter cycle event. When enabled, an event will be generated on begin or end of counter cycle depending of on EVCTRL.CNTSEL(EVCTRL<7:6>) settings.

Value	Description
0	Counter cycle output event is disabled and will not be generated.
1	Counter cycle output event is enabled and will be generated depending on EVCTRL.CNTSEL(EVCTRL<7:6>) value.

Bit 9 – TRGEO Retrigger Output Event Enable

This bit is used to enable the counter retrigger event. When enabled, an event will be generated when the counter retriggers operation.

Value	Description
0	Counter retrigger event is disabled and will not be generated.
1	Counter retrigger event is enabled and will be generated for every counter retrigger.

Bit 8 – OVFE0 Overflow/Underflow Output Event Enable

This bit is used to enable the overflow/underflow event. When enabled an event will be generated when the counter reaches the TOP or the ZERO value.

Value	Description
0	Overflow/underflow counter event is disabled and will not be generated.
1	Overflow/underflow counter event is enabled and will be generated for every counter overflow/underflow.

Bits 7:6 – CNTSEL[1:0] Timer/counter Output Event Mode

These bits define on which part of the counter cycle the counter event output is generated.

Value	Name	Description
0	START	An interrupt/event is generated when a new counter cycle starts
1	END	An interrupt/event is generated when a counter cycle ends
2	BETWEEN	An interrupt/event is generated when a counter cycle ends, except for the first and last cycles
3	BOUNDARY	An interrupt/event is generated when a new counter cycle starts or a counter cycle ends

Bits 5:3 – EVACT1[2:0] Timer/counter Input Event1 Action

These bits define the action the TCC will perform on TCE1 event input.

Value	Name	Description
0	OFF	Event action disabled
1	RETRIGGER	Re-trigger counter on event
2	DIR	Direction control
3	STOP	Stop counter on event
4	DEC	Decrement counter on event
5	PPW	Period capture value in CC0 register, pulse width capture value in CC1 register
6	PWP	Period capture value in CC1 register, pulse width capture value in CC0 register
7	FAULT	Non-recoverable fault

Bits 2:0 – EVACT0[2:0] Timer/counter Input Event0 Action

These bits define the action the TCC will perform on TCE0 event input 0.

Value	Name	Description
0	OFF	Event action disabled
1	RETRIGGER	Start, restart or re-trigger counter on event
2	COUNTEV	Count on event
3	START	Start counter on event
4	INC	Increment counter on event
5	COUNT	Count on active state of asynchronous event
6	STAMP	Stamp capture
7	FAULT	Non-recoverable fault

40.7.11 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x24
Reset: 0x00000000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

Table 40-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	MC[7:0]							
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Reset	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0
Access					ERR	CNT	TRG	OVF
Reset					0	0	0	0

Bits 23:16 – MC[7:0] Match or Capture Channel x Interrupt Enable

Bit 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Recoverable Fault 1 Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault 1 interrupt.

Value	Description
0	The Non-Recoverable Fault 1 interrupt is disabled.
1	The Non-Recoverable Fault 1 interrupt is enabled.

Bit 14 – FAULT0 Non-Recoverable Fault 0 Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Recoverable Fault 0 Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault 0 interrupt.

Value	Description
0	The Non-Recoverable Fault 0 interrupt is disabled.
1	The Non-Recoverable Fault 0 interrupt is enabled.

Bit 13 – **FAULTB** Recoverable Fault B Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – **FAULTA** Recoverable Fault A Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – **DFS** Non-Recoverable Debug Fault Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – **UFS** Non-Recoverable Update Fault Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will clear the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which disables the Non-Recoverable Update Fault interrupt.

Note: This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – **ERR** Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – **CNT** Counter Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG Retrigger Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.7.12 Interrupt Enable Set

Name: INTENSET
Offset: 0x28
Reset: 0x00000000
Property: RW

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

Table 40-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	MC[7:0]							
Reset								
Bit	15	14	13	12	11	10	9	8
Access	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Reset	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0
Access					ERR	CNT	TRG	OVF
Reset					0	0	0	0

Bits 23:16 – MC[7:0] Match or Capture Channel x Interrupt Enable

Bit 15 – FAULT1 Non-Recoverable Fault 1 Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Non-Recoverable Fault 1 Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault 1 interrupt.

Value	Description
0	The Non-Recoverable Fault 1 interrupt is disabled.
1	The Non-Recoverable Fault 1 interrupt is enabled.

Bit 14 – FAULT0 Non-Recoverable Fault 0 Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Non-Recoverable Fault 0 Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault 0 interrupt.

Value	Description
0	The Non-Recoverable Fault 0 interrupt is disabled.
1	The Non-Recoverable Fault 0 interrupt is enabled.

Bit 13 – **FAULTB** Recoverable Fault B Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

Value	Description
0	The Recoverable Fault B interrupt is disabled.
1	The Recoverable Fault B interrupt is enabled.

Bit 12 – **FAULTA** Recoverable Fault A Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

Value	Description
0	The Recoverable Fault A interrupt is disabled.
1	The Recoverable Fault A interrupt is enabled.

Bit 11 – **DFS** Non-Recoverable Debug Fault Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

Value	Description
0	The Debug Fault State interrupt is disabled.
1	The Debug Fault State interrupt is enabled.

Bit 10 – **UFS** Non-Recoverable Update Fault Interrupt Enable

Writing a zero to this bit has no effect.

Writing a one to this bit will set the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which enables the Non-Recoverable Update Fault interrupt.

Note: This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

Value	Description
0	The Non-Recoverable Update Fault interrupt is disabled.
1	The Non-Recoverable Update Fault interrupt is enabled.

Bit 3 – **ERR** Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Error interrupt.

Value	Description
0	The Error interrupt is disabled.
1	The Error interrupt is enabled.

Bit 2 – **CNT** Counter Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Counter interrupt.

Value	Description
0	The Counter interrupt is disabled.
1	The Counter interrupt is enabled.

Bit 1 – TRG Retrigger Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

Value	Description
0	The Retrigger interrupt is disabled.
1	The Retrigger interrupt is enabled.

Bit 0 – OVF Overflow Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enables the Overflow interrupt request.

Value	Description
0	The Overflow interrupt is disabled.
1	The Overflow interrupt is enabled.

40.7.13 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x2C
Reset: 0x00000000
Property: RW

Note: Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 40-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access	MC[7:0]							
Reset								
Bit	15	14	13	12	11	10	9	8
Access	FAULT1	FAULT0	FAULTB	FAULTA	DFS	UFS		
Reset	0	0	0	0	0	0		
Bit	7	6	5	4	3	2	1	0
Access					ERR	CNT	TRG	OVF
Reset					0	0	0	0

Bits 23:16 – MC[7:0] Match or Capture x

Bit 15 – FAULT1 Non-Recoverable Fault 1

This flag is set and resynchronized on the APB clock after a Non-Recoverable Fault 1 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 1 interrupt flag.

Bit 14 – FAULT0 Non-Recoverable Fault 0

This flag is set and resynchronized on the APB clock after a Non-Recoverable Fault 0 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 0 interrupt flag.

Value	Description
0	The Non-Recoverable Fault y interrupt is disabled.
1	The Non-Recoverable Fault y interrupt is enabled.

Bit 13 – FAULTB Recoverable Fault B

This flag is set and resynchronized on the APB clock after a Recoverable Fault B occurs.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 12 – FAULTA Recoverable Fault A

This flag is set and resynchronized on the APB clock after a Recoverable Fault B occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

Bit 11 – DFS Non-Recoverable Debug Fault

This flag is set and resynchronized on the APB clock after an Debug Fault State occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Debug Fault State interrupt flag.

Bit 10 – UFS Non-Recoverable Update Fault

This flag is set when the Ramp index changes and the Lock Update bit is set (CTRLBSET.LUPD(CTRLBSET<1>)).
Writing a zero to this bit has no effect.

Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.

Note: This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

Bit 3 – ERR Error

This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel y interrupt flag is one. In which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the error interrupt flag.

Bit 2 – CNT Counter

This flag is set and resynchronized on the APB clock after a counter event occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CNT interrupt flag.

Bit 1 – TRG Retrigger

This flag is set and resynchronized on the APB clock after a counter retrigger occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the re-trigger interrupt flag.

Bit 0 – OVF Overflow

This flag is set and resynchronized on the APB clock after an overflow condition occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

40.7.14 Status

Name: STATUS
Offset: 0x30
Reset: 0x00000001
Property: RW

Table 40-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CMP[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CCBUFV[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FAULT1	FAULT0	FAULTB	FAULTA	FAULT1IN	FAULT0IN	FAULTBIN	FAULTAIN
Access					R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PERBUFV		PATTBUFV	SLAVE	DFS	UFS	IDX	STOP
Access				R			R	R
Reset	0		0	0	0	0	0	1

Bits 31:24 – CMP[7:0] Compare Channel x Value

Bits 23:16 – CCBUFV[7:0] Compare Channel x Buffer Valid

Bit 15 – FAULT1 Non-Recoverable Fault 1 State

This bit is set by hardware as soon as non-recoverable Fault 1 condition occurs.

This bit is cleared by writing a one to this bit and when the corresponding STATUS.FAULT1IN(STATUS<11>) status bit is low.

Once this bit is cleared, the TCC will restart from the last COUNT value. To restart the TCC from BOTTOM, the TCC restart (RETRIGGER) command (CTRLBSET.CMD (CTRLBSET<7:5>)=1) must be executed before clearing the FAULT1 STATE bit. For further details on TCC commands, refer to the available commands description in the CTRLBSET register.

Bit 14 – FAULT0 Non-Recoverable Fault 0 State

This bit is set by hardware as soon as non-recoverable Fault 0 condition occurs.

This bit is cleared by writing a one to this bit and when the corresponding STATUS.FAULT0IN(STATUS<10>) status bit is low.

Once this bit is cleared, the TCC will restart from the last COUNT value. To restart the TCC from BOTTOM, the TCC restart (RETRIGGER) command (CTRLBSET.CMD (CTRLBSET<7:5>)=1) must be

executed before clearing the FAULT0 STATE bit. For further details on TCC commands, refer to the available commands description in the CTRLBSET register.

Bit 13 – FAULTB Recoverable Fault B State

This bit is set by hardware as soon as recoverable Fault B condition occurs.

This bit can be clear by the hardware when Fault B action is resumed, or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If software halt command is enabled (FCTRLB.HALT (FCTRLB<9:8>) =SW), clearing this bit will release the timer/counter.

Bit 12 – FAULTA Recoverable Fault A State

This bit is set by hardware as soon as recoverable Fault A condition occurs.

This bit can be clear by the hardware when Fault A action is resumed, or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If software halt command is enabled (FCTRLA.HALT (FCTRLA<9:8>) =SW), clearing this bit will release the timer/counter.

Bit 11 – FAULT1IN Non-Recoverable Fault1 Input

This bit is set while an active Non-Recoverable Fault 1 input is present.

Bit 10 – FAULT0IN Non-Recoverable Fault0 Input

This bit is set while an active Non-Recoverable Fault 0 input is present.

Bit 9 – FAULTBIN Recoverable Fault B Input

This bit is set while an active Recoverable Fault B input is present.

Bit 8 – FAULTAIN Recoverable Fault A Input

This bit is set while an active Recoverable Fault A input is present.

Bit 7 – PERBUFV Period Buffer Valid

This bit is set when a new value is written to the PERBUF register. The bit is cleared either by writing a '1' to the corresponding location when CTRLBSET.LUPD (CTRLBSET<1>) is set, or automatically on an UPDATE condition.

Bit 5 – PATTBUFV Pattern Buffer Valid

Bit 4 – SLAVE Slave

Bit 3 – DFS Non-Recoverable Debug Fault State

This bit is set by hardware in Debug mode when DDBGCTRL.FDDBD(DDBGCTRL<2>) bit is set. The bit is cleared by writing a '1' to this bit and when the TCC is not in Debug mode.

When the bit is set, the counter is halted and the Waveforms state depend on DRVCTRL.NREy and DRVCTRL.NRVy registers.

Bit 2 – UFS Non-recoverable Update Fault State

This bit is set by hardware when the Ramp index changes and the Lock Update bit is set (CTRLBSET.LUPD (CTRLBSET<1>)). The bit is cleared by writing a one to this bit.

When the bit is set, the waveforms state depend on DRVCTRL.NREy and DRVCTRL.NRVy registers.

Bit 1 – IDX Ramp

In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads zero. For details on Ramp operations, refer to the .

Bit 0 – STOP Stop

This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when One-Shot operation mode is enabled (CTRLBSET.ONESHOT(CTRLBSET <2>)=1).

This bit is clear on the next incoming counter increment or decrement.

Value	Description
0	Counter is running.
1	Counter is stopped.

40.7.15 Count

Name: COUNT
Offset: 0x34
Reset: 0x00000000
Property: RW

Note: Prior to any read access, this register must be synchronized by user by writing the according TCC Command value to the Control B Set register (CTRLBSET.CMD(CTRLBSET<7:5>)=READSYNC).

Table 40-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	COUNT[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	COUNT[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	COUNT[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	COUNT[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – COUNT[31:0] Counter Value

These bits hold the value of the Counter register.

40.7.16 Pattern

Name: PATT
Offset: 0x38
Reset: 0x0000
Property: RW

Table 40-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PGV[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PGE[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PGV[7:0] Pattern Generator x Output Value

Bits 7:0 – PGE[7:0] Pattern Generator x Output Enable

40.7.17 Waveform Control

Name: WAVE
Offset: 0x3C
Reset: 0x00000000
Property: RW

Table 40-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						SWAP[3:0]		
Access								
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	POL[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
						CICCEN[3:0]		
Access								
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CIPEREN	RAMP[2:0]				WAVEGEN[2:0]		
Access								
Reset	0	0	0	0		0	0	0

Bits 27:24 – SWAP[3:0] Swap DTI Output Pair x

Bits 23:16 – POL[7:0] Channel x Polarity

Bits 11:8 – CICCEN[3:0] Circular Channel x Enable

Bit 7 – CIPEREN Circular period Enable

Setting this bits enable the period circular buffer option. When the bit is set, the PER register value is copied-back into the PERBUF register on UPDATE condition.

Bits 6:4 – RAMP[2:0] Ramp Mode

These bits select Ramp operation (RAMP). These bits are not synchronized.

Value	Name	Description
0	RAMP1	RAMP1 operation
1	RAMP2A	Alternative RAMP2 operation
2	RAMP2	RAMP2 operation
3	RAMP2C	Critical RAMP2 operation
4	RAMP2CS	Critical Swapped RAMP2 operation

Bits 2:0 – WAVEGEN[2:0] Waveform Generation

These bits select the waveform generation operation. The settings impact the top value and control if frequency or PWM waveform generation should be used. These bits are not synchronized.

Value	Name	Description	Operation	Top	Update	Waveform Output On Match	Waveform Output On Update	OVF Interrupt Flag/Event Up Down	
0x0	NFRQ	Normal Frequency	PER	TOP/Zero	Toggle	Stable	TOP	Zero	
0x1	MFRQ	Match Frequency	CC0	TOP/Zero	Toggle	Stable	TOP	Zero	
0x2	NPWM	Normal PWM	PER	TOP/Zero	Set	Clear	TOP	Zero	
0x3	DPWM	Dual Compare PWM	PER	TOP/ZERO	Set/Clear	Clear	-	Zero	
0x4	DSCRITICAL	Dual-slope PWM	PER	Zero	~DIR	Stable	-	Zero	
0x5	DSBOTTOM	Dual-slope PWM	PER	Zero	~DIR	Stable	-	Zero	
0x6	DSBOTH	Dual-slope PWM	PER	TOP & Zero	~DIR	Stable	TOP	Zero	
0x7	DSTOP	Dual-slope PWM	PER	Zero	~DIR	Stable	TOP	-	

Value	Name	Description
0	NFRQ	Normal frequency
1	MFRQ	Match frequency
2	NPWM	Normal PWM
3	DPWM	Dual compare PWM
4	DSCRITICAL	Dual-slope critical
5	DSBOTTOM	Dual-slope with interrupt/event condition when COUNT reaches ZERO
6	DSBOTH	Dual-slope with interrupt/event condition when COUNT reaches ZERO or TOP
7	DSTOP	Dual-slope with interrupt/event condition when COUNT reaches TOP

40.7.18 Period

Name: PER
Offset: 0x40
Reset: 0xFFFFFFFF
Property: RW

Table 40-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PER[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	PER[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	PER[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	PER[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – PER[31:0] Period Value

40.7.19 Compare and Capture

Name: CC
Offset: 0x44 + n*0x04 [n=0..3]
Reset: 0x00000000
Property: RW

The CCy register represents the 32-bit value. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output from the comparator is then used for generating waveforms.

CCy register is updated with the buffer value from their corresponding CCBUFy register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

Table 40-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CC[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CC[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CC[31:0] Channel Compare/Capture Value

These bits hold the value of the Channel y compare/capture register.

40.7.20 Pattern Buffer

Name: PATTBUF
Offset: 0x64
Reset: 0x0000
Property: RW

Table 40-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	15	14	13	12	11	10	9	8
	PGVB[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PGEb[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 15:8 – PGVB[7:0] Pattern Generation x Output Value Buffer

Bits 7:0 – PGEb[7:0] Pattern Generation x Output Enable Buffer

40.7.21 Period Buffer

Name: PERBUF
Offset: 0x6C
Reset: 0xFFFFFFFF
Property: RW

Table 40-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PERBUF[31:24]							
Access								
Reset	1	1	1	1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
	PERBUF[23:16]							
Access								
Reset	1	1	1	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8
	PERBUF[15:8]							
Access								
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	PERBUF[7:0]							
Access								
Reset	1	1	1	1	1	1	1	1

Bits 31:0 – PERBUF[31:0] Period Buffer Value

These bits hold the value of the Period Buffer register. The value is copied to PER register on UPDATE condition.

40.7.22 Compare and Capture Buffer

Name: CCBUF
Offset: $0x70 + n \cdot 0x04$ [$n=0..3$]
Reset: 0x00000000
Property: RW

CCBUF_y is copied into CC_y at TCC update time.

Table 40-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CCBUF[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	CCBUF[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	CCBUF[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	CCBUF[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CCBUF[31:0] Channel Compare/Capture Buffer Value

These bits hold the value of the Channel y Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CC_y). Accessing this register using the CPU or DMA will affect the corresponding STATUS.CCBUF_{Vy} status bit.

41. TrustRAM (TRAM)

41.1 Overview

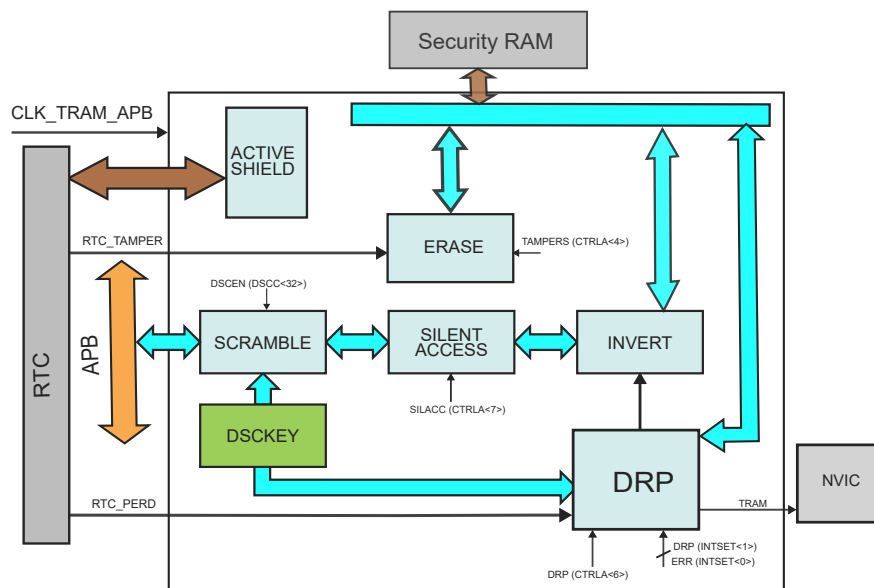
The TrustRAM (TRAM) module is the controller interface for 512 Bytes of security RAM. This RAM is intended for volatile secret data.

41.2 Features

- Address scrambling to the security RAM
- Data scrambling to the security RAM and data descrambling from the security RAM
- Data remanence prevention
- Active shielding to prevent physical tamper on security RAM
- Full erasure of scramble key and security RAM data on tamper detection

41.3 Block Diagram

Figure 41-1. Block Diagram



41.4 Functional Description

41.4.1 Principle of Operation

System bus transactions from the CPU to the security RAM undergo a scrambling routine. The TRAM module modifies both address and data bus information through an algorithm determined by a scrambling key (DSCC.DSCKEY (DSCC<29:0>)). This is performed on both write and read transactions. When the TRAM module receives a Tamper Event (RTC_TAMPER) from the RTC module, it erases the full security RAM and the scrambling key. When it receives a RTC Interval Periodic Event (RTC_PERD) from the RTC module, the TRAM module runs a data remanence routine on the security RAM. The TRAM module can be configured to generate interrupts. The following sections describe each operation in detail.

41.4.2 Basic Operation

41.4.2.1 Clocks

The TRAM bus clock (CLK_TRAM_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRAM_APB can be found in the *Peripheral Clock Masking* section.

41.4.2.2 Register Access Protection

All registers with write-access are optionally write-protected in the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag (INTFLAG) register
- All RAM addresses

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the *Peripheral Access Controller* chapter for details.

41.4.2.3 Enable Protection

The following bits and registers are enable-protected, meaning that they can only be written when the TRAM module is disabled (CTRLA.ENABLE bit (CTRLA <1>) = '0'):

- Tamper Erase bit in the Control A register (CTRLA.TAMPERS bit (CTRLA <4>))
- Data Remanence Protection bit in the Control A register (CTRLA.DRP bit (CTRLA <6>))
- Data Scramble Control register (DSCC) register

When the CTRLA.ENABLE bit (CTRLA <1>) is '0', enable-protected bits in the CTRLA register can be written at the same time the CTRLA.ENABLE bit (CTRLA <1>) is written to '1'. However, when the CTRLA.ENABLE bit (CTRLA <1>) is '1' these bits can not be written at the same time the CTRLA.ENABLE bit (CTRLA <1>) is written. Enable-protection is denoted by the Enable-Protected property in the register description.

41.4.2.4 Enabling, Disabling and Resetting

The TRAM is enabled by writing a '1' to the Enable bit in the Control A register CTRLA.ENABLE bit (CTRLA <1>). The TRAM is disabled by writing a '0' to CTRLA.ENABLE bit (CTRLA <1>).

The TRAM is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST bit (CTRLA <0>)). All registers in the TRAM will be reset to their initial state, and the TRAM will be disabled. All data in the security RAM will be cleared to '0'.

41.4.2.5 Synchronization

In the TRAM module some bits must be synchronized when accessed. A register or register bits may require:

- Synchronization when written
- No synchronization

When executing an operation that requires synchronization, the corresponding status bit in the Synchronization Busy (SYNCBUSY) register sets immediately, and is cleared when synchronization is complete.

If an operation that requires the execution of synchronization while the corresponding bit in SYNCBUSY register is set to '1', the operation is discarded and an error is generated. In the TRAM module, the following bits need synchronization when written:

- Software Reset bit in Control A register (CTRLA.SWRST bit (CTRLA <0>))
- Enable bit in Control A register (CTRLA.ENABLE bit (CTRLA <1>))

41.4.2.6 Initialization

The following steps must be performed to operate the TRAM module in basic mode:

- Configure the clock source for CLK_TRAM_APB in the Main Clock Controller (MCLK) and enable the clock by writing a '1' to the TRAM bit in the APB MCLK.CLKMSK Mask register
- Clear all registers by writing '1' to CTRLA.SWRST bit (CTRLA<0>) and wait for the SYNCBUSY.SWRST bit (SYNCBUSY<0>) to set
- Program data scrambling key in DSCC.DSCKEY bits (DSCC <29:0>) and enable scrambling by programming DSCEN bit (DSCC <31>)
- Enable the TRAM module by setting CTRLA.ENABLE bit (CTRLA<1>) and wait for the SYNCBUSY.CTRLA bit (SYNCBUSY<1>) to set

41.4.2.7 Scrambling

The Data Scramble Control (DSCC) must be configured before the CTRLA.ENABLE (CTRLA <1>) is set. These settings cannot be changed while the module is enabled.

The scrambling logic is enabled by writing '1' to the enable bit in the Data Scramble Control register (DSCC.DSCEN bit (DSCC <31>)). Scrambling is disabled by writing a '0' to DSCC.DSCEN bit (DSCC <31>).

41.4.2.8 Silent Access

Silent access bit (CTRLA.SILACC (CTRLA <7>)) must be configured before CTRLA.ENABLE bit (CTRLA <1>) is set. This setting cannot be changed while the module is enabled. When this mode is enabled, only half of the security RAM (4KB) is available for data storage since the other half is reserved to store the 1's complement (bitwise invert) values.

The TRAM module executes the following protocols:

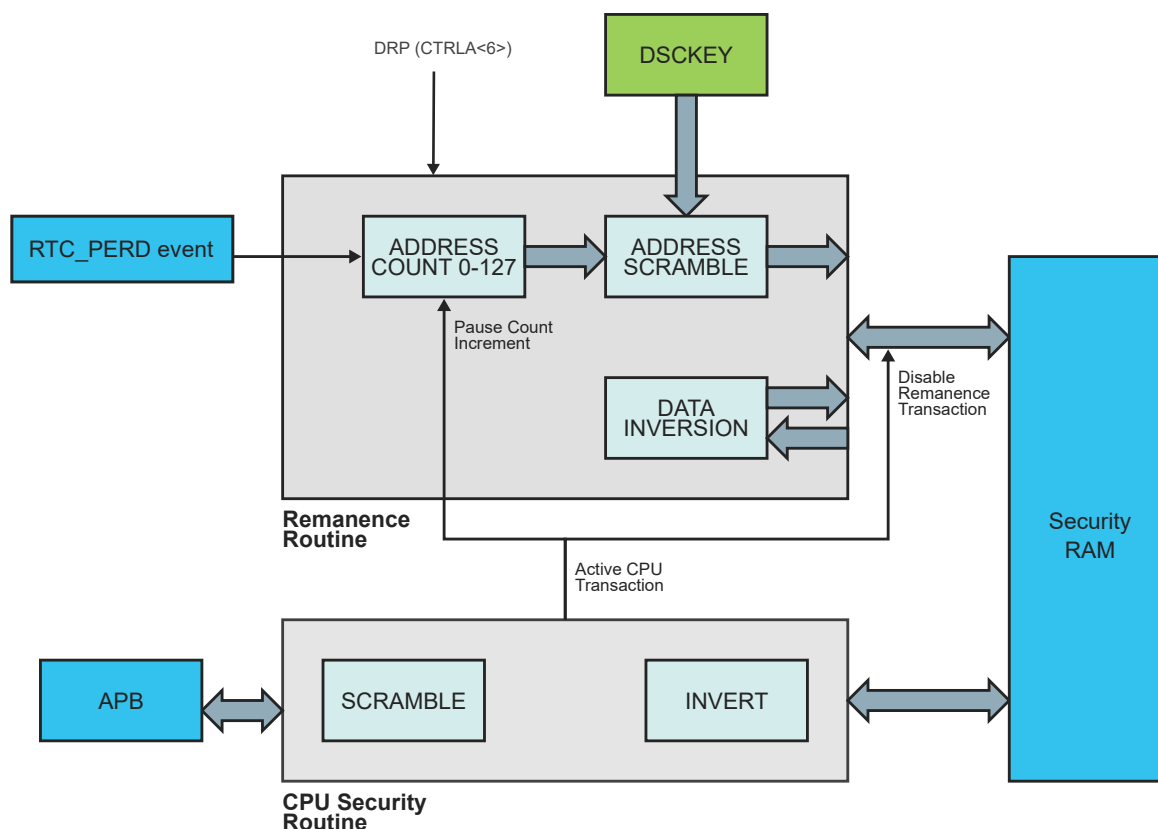
- When the CPU writes to the security RAM, the data and its bitwise invert are stored into the security RAM.
- When the CPU reads from the security RAM, both the data and its bitwise invert are retrieved from the security RAM. If the TRAM module cannot verify that both values complement each other, a bus error is returned.

41.4.2.9 Data Remanence Prevention

Data remanence prevention can be enabled by setting the CTRLA.DRP (CTRLA <6>) bit. The Data remanence prevention bit (CTRLA.DRP (CTRLA <6>)) must be configured before the CTRLA.ENABLE bit (CTRLA <1>) is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Periodic Interval Event (RTC_PERD) will trigger the automated data remanence routine. An internal counter will count from 0 to 127 and serves as the address access bus to the security RAM. For every address iteration, the TRAM module reads the word data from the security RAM, inverts the value and writes back to the same address. To prevent linear access to the security RAM, the remanence address value is scrambled using the same protocols as a CPU address scramble. After remanence has updated all address locations, the routine will end by toggling the RAM inversion status bit (RAMINV bit (STATUS <0>)). See the following figure.

Data remanence is a low-priority routine. If the CPU attempts to access the security RAM while remanence is active, the routine is temporarily paused until the CPU access is completed. If a tamper full erase event is detected, the remanence routine is aborted and the internal address counter will reset to 0.

Figure 41-2. Remanence Routine



Software should erase or overwrite sensitive information from the TRAM before the system enters the OFF state in order to reduce the chances of any data remanence.

41.4.2.10 Tamper Full Erase

Tamper full erase bit (CTRLA.TAMPERS (CTRLA <4>)) must be configured before CTRLA.ENABLE bit (CTRLA <1>) is set. This setting cannot be changed while the module is enabled. When this feature is enabled, the RTC Tamper Event (RTC_TAMPER) will trigger the full erase equivalent to a TRAM module software reset and the reset of the Data Scramble Key (DSCC.DSCKEY bits (DSCC <29:0>)) register. All the TRAM registers are reverted to the default reset value. Data inside the security RAM is written to '0' for all address locations.

The tamper full erase routine operates at the highest priority. If a remanence routine executing when a tamper full erase occurs, the remanence routine is immediately terminated. If the CPU attempts to write a new scramble key at the same time the tamper key erase routine is active, the CPU data is ignored, but no bus error will occur. If a CPU security routine access is requested during a tamper full erase, the CPU transaction will be ignored and treated as a bus error similar to accessing the module during a software reset.

Note: When the power mode is set to a Low voltage mode, the voltage must be ramped back up by the RPMU /SUPC to nominal VDDCORE levels before the TRAM erase can be performed in response to a tamper detection.

41.4.3 Interrupts

The TRAM module has the following interrupt sources:

- Data Remanence Prevention (DRP): Indicates that the data remanence prevention routine has ended.

- Data Read Error (ERR): Indicates when there is a RAM readout error.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the TRAM is reset. See INTFLAG for details on how to clear interrupt flags. The interrupt request lines are connected to the interrupt controller. Both interrupt requests from the TRAM module are ORed together on the system level to generate one combined interrupt request to the NVIC. This combined interrupt is called the TRAM interrupt. Using the TRAM module interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

The following additional steps are needed for the TRAM module to operate in the interrupt mode before enabling the TRAM module:

- Configure the NVIC by setting group priority, sub priority and by enabling TRAM IRQ
- Enable TRAM Data Read Error (ERR) interrupt by setting INTENSET.ERR bit (INTENSET<0>)
- Enable TRAM Data Remanence Prevention (DRP) interrupt by setting INTENSET.DRP bit (INTENSET<1>)
- In the TRAM Interrupt Service Routine write '1' in INTFLAG.DRP bit (INTFLAG <1>) to clear the DRP interrupt and write '1' in INTFLAG.ERR bit (INTFLAG <0>) to clear the ERR interrupt

41.4.4 Events

The RTC Period Interval event (RTC_PERD) for Data Remanence Prevention and the RTC Tamper (RTC_TAMPER) event are directly connected from the RTC to the TRAM, without going through the Event System. No output events are connected from the TRAM module to Event Systems.

41.4.5 Sleep Mode Operation

The TRAM continues to operate during sleep. When it receives events from the RTC, it will request its own clock in order to perform the requested operation (remanence and tamper erase).

The TRAM will continue to operate in any sleep mode, as long as its source clock is running. The TRAM interrupt can be used to wake up the device from sleep mode.

41.5 Peripheral Dependencies

Table 41-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
TRAM	0x4500A000	33 : ERR/DRP	APB: CLKMSK[0] MASK15	---	15 STATUS0 PERID15			PD_CORE_BU

41.6 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0	SILACC	DRP		TAMPERS			ENABLE	SWRST
0x04	INTENCLR	31:24								
		23:16								
		15:8								
		7:0							DRP	ERR
0x08	INTENSET	31:24								
		23:16								
		15:8								
		7:0							DRP	ERR
0x0C	INTFLAG	31:24								
		23:16								
		15:8								
		7:0							DRP	ERR
0x10	STATUS	31:24								
		23:16								
		15:8								
		7:0							DRP	RAMINV
0x14	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x18	DSCC	31:24	DSCEN		DSCKEY[29:24]					
		23:16			DSCKEY[23:16]					
		15:8			DSCKEY[15:8]					
		7:0			DSCKEY[7:0]					
0x1C	PERMW	31:24								
		23:16								
		15:8								
		7:0						DATA[2:0]		
0x20	PERMR	31:24								
		23:16								
		15:8								
		7:0						DATA[2:0]		
0x24 ... 0x0FFF	Reserved									
0x1000	WORD0	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							
...										
0x10FC	WORD63	31:24	DATA[31:24]							
		23:16	DATA[23:16]							
		15:8	DATA[15:8]							
		7:0	DATA[7:0]							

41.6.1 Control

Name: CTRLA
Offset: 0x0000
Reset: 0x00000000
Property: RW

Notes:

1. Only ENABLE and SWRST bits are Write Synchronized.
2. Read/Write access to this register is limited to 32-bit width. Byte level access is not allowed.
3. Reserved bits must always be written as '0'.
4. TRAM module registers are not reset during a soft system reset.

Table 41-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	SILACC	DRP		TAMPERS			ENABLE	SWRST
Access	RW	RW		RW			RW	W
Reset	0	0		0			0	0

Bit 7 – SILACC Silent Access

Enables differential storage of data.

Value	Description
0	Silent access is disabled.
1	Silent access is enabled.

Bit 6 – DRP Data Remanence Prevention

Enables periodic Data remanence prevention (DRP) in the TRAM module.

Value	Description
0	Data remanence prevention is disabled.
1	Data remanence prevention is enabled.

Bit 4 – TAMPERS Tamper Erase

This bit enables auto-erase of the security RAM and DSCKEY bits (DSCC <29:0) on tamper event.

Value	Description
0	Tamper erase is disabled.
1	Tamper erase is enabled.

Bit 1 – ENABLE Enable

This bit is not Enable-Protected.

Value	Description
0	The TRAM module is disabled.
1	The TRAM module is enabled.

Bit 0 – SWRST Software Reset

Writing a '0' to this bit has no effect.

Writing a '1' to this bit resets all registers in the TRAM module to their initial state, and the TRAM module will be disabled. This bit can also be set via hardware when a tamper occurs while CTRLA.TAMPERS bit (CTRLA <4>) is set.

Writing a one to CTRLA.SWRST bit (CTRLA <0>) will always take precedence, meaning that all other writes in the same write operation will be discarded.

Due to synchronization there is a delay from writing CTRLA.SWRST bit (CTRLA <0>) until the reset is complete. CTRLA.SWRST bit (CTRLA <0>) and SYNCBUSY.SWRST bit (SYNCBUSY <0>) will both be cleared when the reset is complete.

Reading this bit provides the following information.

Notes:

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by hardware.

Value	Description
0	The reset operation is not ongoing.
1	The reset operation is ongoing.

41.6.2 Interrupt Enable Clear

Name: INTENCLR
Offset: 0x0004
Reset: 0x00000000
Property: RW

Notes:

- Access to this register is limited to 32-bit width. Byte level access is not allowed.
- Reserved bits must always be written as '0'.

Table 41-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DRP	ERR
Access							RW	RW
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Ended Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Data Remanence Prevention Complete Interrupt Enable bit, which disables the data remanence prevention complete interrupt. Reading this bit provides the following information.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Readout Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the TRAM Read Error Interrupt Enable bit, which disables the TRAM read error interrupt. Reading this bit provides the following information.

Value	Description
0	TRAM read error interrupt is disabled.

Value	Description
1	TRAM read error interrupt is enabled.

41.6.3 Interrupt Enable Set

Name: INTENSET
Offset: 0x0008
Reset: 0x00000000
Property: RW

Notes:

1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as '0'.

Table 41-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DRP	ERR
Access							RW	RW
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Ended Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the Data Remanence Prevention Complete Interrupt Enable bit, which enables the data remanence prevention complete interrupt. Reading this bit provides the following information.

Value	Description
0	Data remanence prevention complete interrupt is disabled.
1	Data remanence prevention complete interrupt is enabled.

Bit 0 – ERR TrustRAM Readout Error Interrupt Enable

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will set the TRAM Read Error Interrupt Enable bit, which enables the TRAM read error interrupt. Reading this bit provides the following information.

Value	Description
0	TRAM read error interrupt is disabled.

Value	Description
1	TRAM read error interrupt is enabled.

41.6.4 Interrupt Flag Status and Clear

Name: INTFLAG
Offset: 0x000C
Reset: 0x00000000
Property: RW

Notes:

1. Access to this register is limited to 32-bit width. Byte level access is not allowed.
2. Reserved bits must always be written as '0'.
3. Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

Table 41-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DRP	ERR
Access							RW	RW
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Ended

This flag is set when the data remanence prevention routine has completed, and an interrupt request will be generated if INTENSET.DRP bit (INTENSET <1>) is set to '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the data remanence prevention complete interrupt flag.

Bit 0 – ERR TrustRAM Readout Error

This flag is set when an error is detected in the TRAM readout, and an interrupt request will be generated if INTENSET.ERR bit (INTENSET <0>) is set to '1'.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit clears the TRAM read error interrupt flag.

41.6.5 Status

Name: STATUS
Offset: 0x0010
Reset: 0x00000000
Property: R

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 41-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							DRP	RAMINV
Access							RW	R
Reset							0	0

Bit 1 – DRP Data Remanence Prevention Ongoing

This bit provides the status of the data remanence prevention routine.

Value	Description
0	The data remanence prevention routine is not running.
1	The data remanence prevention routine is running.

Bit 0 – RAMINV RAM Inversion Bit

This bit provides the status of the TRAM bit values inversion function.

Value	Description
0	The TRAM physical bit information is normal.
1	The TRAM physical bit information is inverted.

41.6.6 Synchronization Busy Status

Name: SYNCBUSY
Offset: 0x0014
Reset: 0x00000000
Property: R

Note: Access to this register is limited to 32-bit width. Byte level access is not allowed.

Table 41-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R	R
Reset							0	0

Bit 1 – ENABLE Enable Busy

Value	Description
0	Write synchronization for CTRLA.ENABLE bit (CTRLA <1>) bit is complete.
1	Write synchronization for CTRLA.ENABLE bit (CTRLA <1>) bit is ongoing.

Bit 0 – SWRST Software Reset Busy

This bit will set in two ways:

- Writing '1' to CTRLA.SWRST bit (CTRLA <0>)
- A tamper event occurs when CTRLA.TAMPERS (CTRLA <4>) = '1' (i.e., Tamper erase is enabled)

Value	Description
0	Write synchronization for CTRLA.SWRST bit (CTRLA <0>) bit is complete.
1	Write synchronization for CTRLA.SWRST bit (CTRLA <0>) bit is ongoing.

41.6.7 Data Scramble Control

Name: DSCC
Offset: 0x0018
Reset: 0x00000000
Property: W

Notes:

- Access to this register is limited to 32-bit width. Byte level access is not allowed.
- Reserved bits must always be written as '0'.

Table 41-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DSCEN		DSCKEY[29:24]					
Access	RW		RW	RW	RW	RW	RW	RW
Reset	0		0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DSCKEY[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DSCKEY[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DSCKEY[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 31 – DSCEN Data Scramble Enable

Writing '1' to this bit enables the TRAM scrambling function. Reading this bit provides the following information:

Value	Description
0	TRAM scrambling function is disabled
1	TRAM scrambling function is enabled

Bits 29:0 – DSCKEY[29:0] Data Scramble Key

The key value used for address and data scrambling and descrambling. Any value written to this field is XOR'ed with the previous data. Writing '1' to CTRLA.SWRST bit (CTRLA <0>) will reset this field to 0. These bits will always return '0' when read.

41.6.8 Permutation Write

Name: PERMW
Offset: 0x001C
Reset: 0x00000000
Property: W

Table 41-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						DATA[2:0]		
Access						RW	RW	RW
Reset						0	0	0

Bits 2:0 - DATA[2:0] Permutation Scrambler Data Input

41.6.9 Permutation Read

Name: PERMR
Offset: 0x0020
Reset: 0x00000000
Property: R

Table 41-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
						DATA[2:0]		
Access						RW	RW	RW
Reset						0	0	0

Bits 2:0 - DATA[2:0] Permutation Scrambler Data Output

41.6.10 TrustRAM

Name: WORD
Offset: 0x1000 + n*0x04 [n=0..63]
Reset: 0x00000000
Property: RW

Table 41-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DATA[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DATA[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DATA[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DATA[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DATA[31:0] Trust RAM Data

42. Peripheral Touch Controller (PTC)

42.1 Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self and mutual-capacitance sensors.

In mutual-capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

The number of available pins and the assignment of X-line and Y-line are depending on both package type and device configuration. Refer to the *Configuration Summary and I/O Multiplexing* table for additional information.



Important: If $AVDD < 2.5V$, users must enable analog charge pumps in the SUPC.VREGCTRL.CPEN[1:0].

42.2 Features

The following are key features of the Peripheral Touch Controller module:

- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels
- Supports wake-up on touch from Standby Sleep mode
- Supports mutual-capacitance and self-capacitance sensing
 - Mix-and-match mutual and self-capacitance sensors
- One pin per electrode, no external components
- Load compensating charge sensing
 - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero drift over the temperature and VDD range
 - Auto calibration and recalibration of sensors
- Single-shot charge measurement
- Hardware noise filtering and noise signal desynchronization for high conducted immunity
- Polarity control, allowing Parallel Acquisition (through the QTouch Library) individually controls the polarity of each line
- Driven Shield+ for better noise immunity and moisture tolerance
 - Any PTC X/Y line can be used for the driven shield
 - All enabled sensors will be driven at the same potential as the sensor scanned
- Selectable channel change delay allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete

42.3 Block Diagram

Figure 42-1. PTC Block Diagram Mutual Capacitance

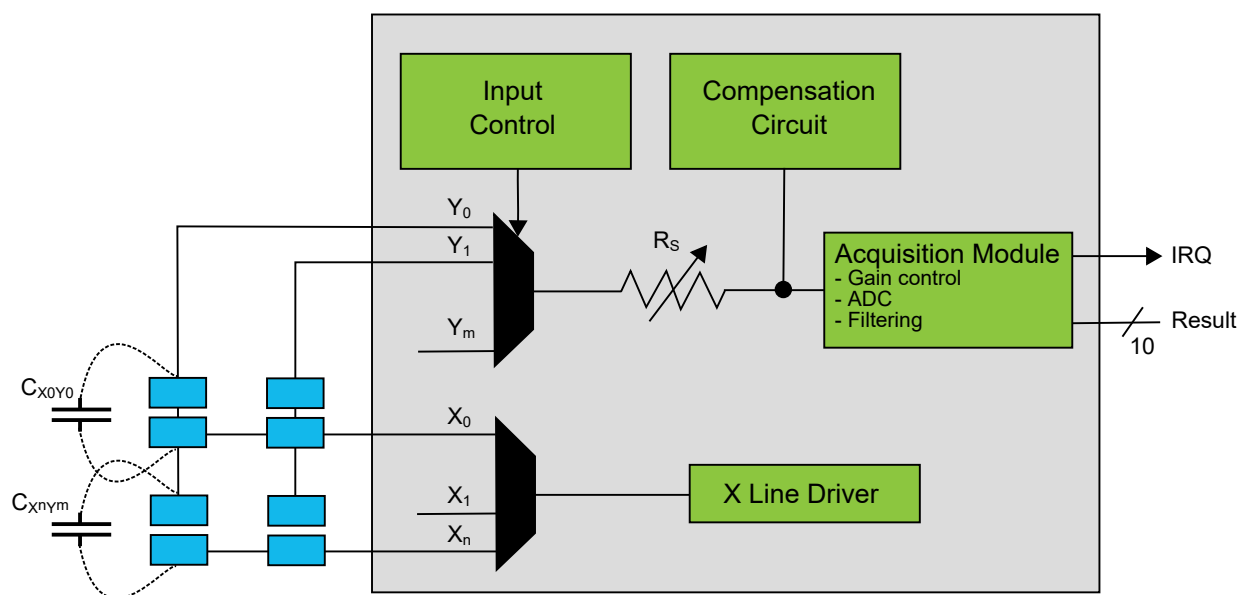
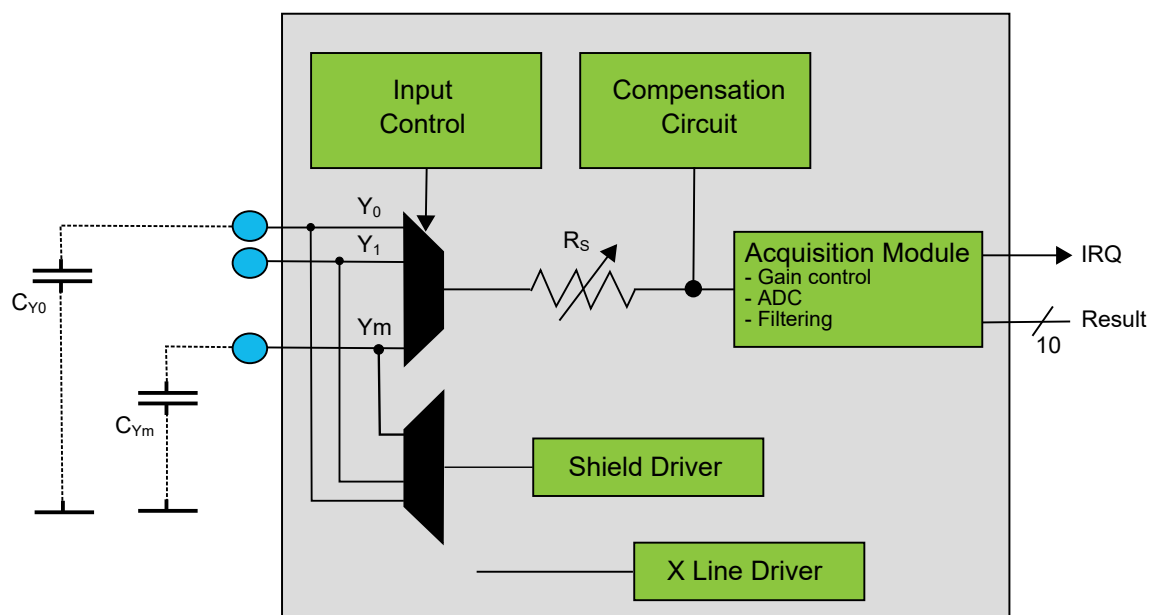


Figure 42-2. PTC Block Diagram Self Capacitance



42.4 Signal Description

Table 42-1. Signal Description for PTC

Name	Type	Description
Y[m:0]	Analog	Y-line (Input/Output)

Table 42-1. Signal Description for PTC (continued)

Name	Type	Description
X[n:0]	Digital	X-line (Output)

Note: The number of X-lines and Y-lines are device dependent. Refer to the [Configuration Summary](#) for additional information.

Refer to the Pinout section for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

42.5 Peripheral Dependencies

In order to use this peripheral, configure the other components of the system as described in the following sections.

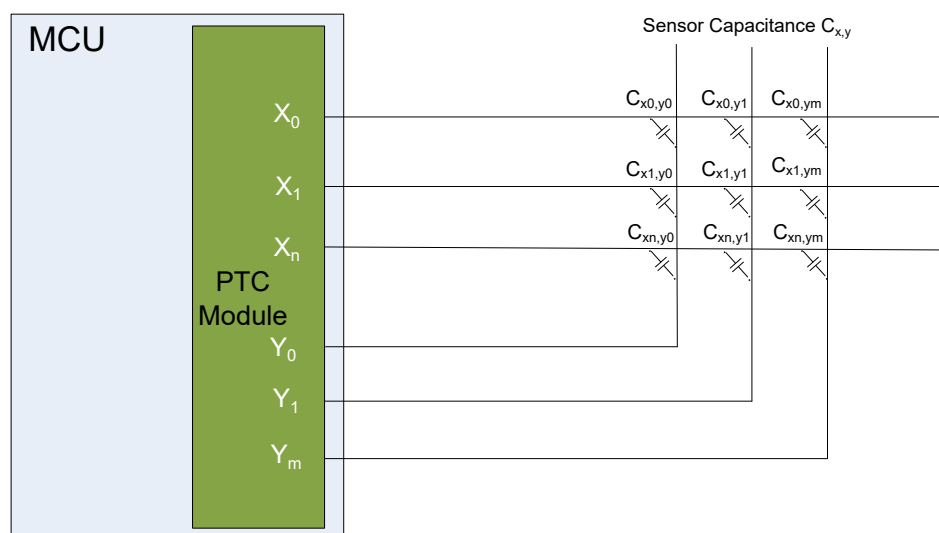
42.5.1 I/O Lines

The I/O lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1kΩ or more can be used on X-lines and Y-lines.

42.5.1.1 Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the Peripheral Touch Controller.

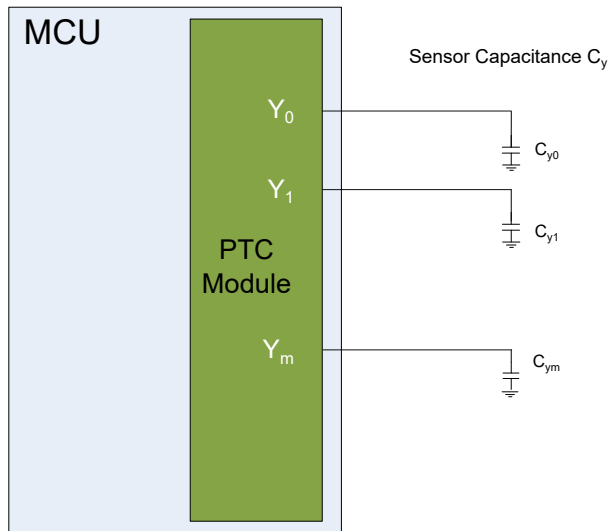
Figure 42-3. Mutual Capacitance Sensor Arrangement



42.5.1.2 Self-Capacitance Sensor Arrangement

A self-capacitance sensor is connected to a single pin on the Peripheral Touch Controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the Peripheral Touch Controller.

Figure 42-4. Self-Capacitance Sensor Arrangement



For more information about designing the touch sensor, refer to [Buttons, Sliders and Wheels Touch Sensor Design Guide](#).

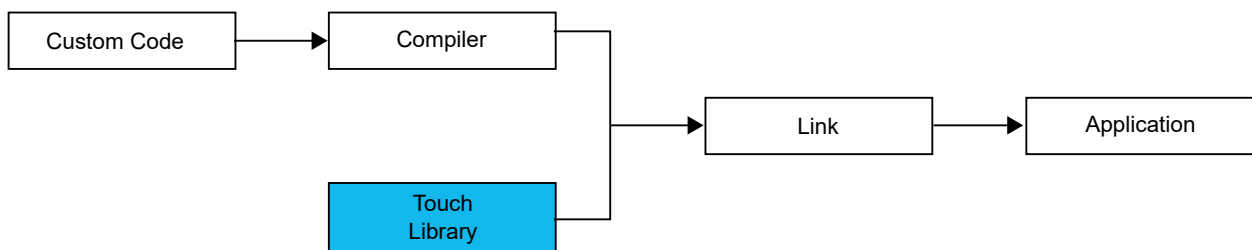
42.5.2 Clocks

The PTC is clocked by the GCLK_PTC clock. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_PTC_APB enabled by default in the MCLK.CLKMSK3.MSK10 register). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to *GCLK - Generic Clock Controller* chapter.

42.6 Functional Description

In order to access the PTC, the user must use the Touch Configurator to configure and link the Touch Library firmware with the application software. Touch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

Figure 42-5. Touch Library Usage



For more information about Touch Library, refer to the Touch Library Peripheral Touch Controller User Guide.

42.6.1 Principle of Operation

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device.

The PTC supports both self and mutual-capacitance sensors.

In mutual-capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

42.6.2 Basic Operation

42.6.2.1 Initialization

The following registers are enable-protected, meaning that they can only be written when the PTC is disabled (CTRLA.ENABLE=0):

- Control A (CTRLA), except ENABLE and SWRSTbits
- Event Control register(EVCTRL)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

42.6.2.2 Enabling, Disabling and Resetting

The PTC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The PTC is disabled by writing CTRLA.ENABLE=0.

To go to the active mode from the off mode, the application must wait for the bias generators to stabilize, this wait is specified in powers of 2 by the exponent bit field CTRLA.WKUPEXP[3:0]. The bit field CTRLA.WKUPEXP[3:0] generates the full count 2WKUPEXP, (where WKUPEXP is mapped internally to be always 4 or greater), which controls how many PTC core clocks are needed in the PTC core to fully power-up the analog bias circuitry. This delay is enforced every time CTRLA.ENABLE has a low to high transition. During this delay, the triggers are blocked from starting conversions, because the analog core is not ready yet. At the end of the delay, the corresponding status bit ACRRDY is set in the INTFLAG register for the user to poll and determine when the PTC Core is ready for operation and the triggers are allowed to start conversions. Because the delays are significant, the ACRRDY status bits can be enabled as part of an group interrupt request.

The PTC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the PTC, except DBGCTRL, will be reset to their initial state, and the PTC will be disabled. Refer to CTRLA for details.

42.6.2.3 Mutual Capacitance Mode

The I/O Port pins that are chosen for touch sensing channels must be set to Analog input/output mode using the Port control signals. The chosen X- and Y-channel set must be enabled for PTC use by writing the desired configuration into I/O Port control registers.

42.6.2.4 Self Capacitance Mode

The I/O Port pins that are chosen for touch sensing channels must be set to Analog input/output mode using the Port control signals. The chosen Y-channel set must be enabled for PTC use by writing the desired configuration into I/O Port control registers.

42.6.3 Driven Shield

Driven shield is the capability to drive an output line in synchronization with TOUCH input. This will provide better noise immunity from noise sources beneath the sensor, and improved sensitivity of the touch signal. Driven shield will only be used in self-cap mode.

42.6.4 Interrupts

The PTC has the following interrupt sources:

- End of Conversion:EOC
- Window Monitor: WCOMP
- Analog Core Ready:ACRRDY

These interrupts are asynchronous wake-up sources. Refer to the Sleep Mode Controller for details.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to the *Nested Vector Interrupt Controller* for details.

42.6.5 Events

The PTC can generate the following output events:

- End Of Conversion (EOC): Generated when the conversion is complete and the result is available. Refer to EVCTRL for details.
- Window Monitor (WCOMP): Generated when the window monitor condition match.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The PTC can take the following actions on an input event:

- Start a conversion
- Restart DMA sequencing

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

The PTC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the PTC will detect a rising edge on the incoming event. If the PTC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

42.7 Peripheral Dependencies

Table 42-2. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
PTC	0x4482C000	133 : EOC/ WCOMP/ ACRRDY	APB: CLKMSK[1] MASK19	PCHCTRL[35]	41 STATUS1 PERID9	38 : EOC 39 : SEQ 40 : WCOMP	<u>Generators:</u> EOC: CHANNEL[81] WCOMP: CHANNEL[82] <u>Users:</u> DSEQR: USER[51],Async STCONV: USER[52],Async	PD_CORE_SW

43. Hardware Security Module Lite (HSM-Lite)

43.1 Overview

The Hardware Security Module Lite (HSM-Lite) peripheral integrates high-performance cryptographic accelerators, a True Random Number Generator (TRNG), and scatter-gather DMA into a single module. The AHB Host port provides access to programming model registers. The AHB Host port is used for high bandwidth access to system memory for encryption, hashing, and public key operations.

In devices with TrustZone, typically the HSM-Lite is configured to be a secure peripheral. This restricts access for the HSM-Lite registers to the secure world only. Secure world firmware provides cryptographic services to the non-secure world using an API. The HSM-Lite has the capability to act as a non-secure Host for these operations.

43.2 Features

The following are key features of the Hardware Security Module Lite:

- AES-128, AES-192, and AES-256: Fully compliant with NIST FIPS 197
 - ECB, CBC, CFB, OFB, CTR, GCM, CCM, XTS, CMAC Modes
- HASH/MAC
 - MD-5, SHA-1, SHA-256, SHA-224, SHA-384 and SHA-512, and SHA3 (SW)
- Key Derivation Function (HKDF, KDF2...)
 - Public Key Cryptography: RSA, DSA, EdDSA, and ECC
 - RSA with or without Chinese Remainder Theorem (CRT). Up to 4096-bit Key length.
- ECC with ECC-GF(p), ECC-GF(2m), and ECDSA support
 - Prime Field P-192, P-224, P-256, P-384, P-521
 - Binary Field K-163, K-233, K-283, K-409, K-571
 - Binary Field B-163, B-233, B-283, B-409, B-571
 - P-224, P-256, P-384, and P-521 Elliptic Curve – ECDSA Sign/Verify
- DSA support up to 2048-bit Key Length
- NIST-800-22 and NIST-800-90B compliant True Random Number Generator (TRNG, AIS-31 Compliant)
- Two 4Kb Key Storage Pages

43.3 Operation

43.3.1 Clock

A single clock running at the system bus frequency is used for all internal logic and both AHB interfaces.

The CTRLA.ENABLE and CTRLA.RUNSTDBY bits control operation of the clock. When CTRLA.ENABLE = 1 the clock is enabled in Active and Idle sleep modes. When CTRLA.ENABLE = 1 and CTRLA.RUNSTDBY = 1 the clock is also requested in standby sleep mode.

43.3.2 Interrupts

HSM-Lite has three interrupt requests. Interrupt processing functions are included in the Microchip Harmony Libraries.

43.3.3 Access Control

The CTRLA.PRIV bit controls access to the HSM-Lite registers. When CTRLA.PRIV = 1, only privileged accesses are permitted to the HSM-Lite registers, unprivileged accesses generate a bus error. When CTRLA.PRIV = 0 both privileged and unprivileged accesses are permitted.

43.3.4 Write Protect

The CTRLA and WPCTRL registers can be write locked. This is controlled through the WPCTRL.WPEN and WPCTRL.WPLCK bits.

43.4 Cryptographic Accelerators

43.4.1 AES

The AES accelerator supports 128, 192 and 256 bit key sizes and the following modes of operation: ECB, CBC, OFB, CFB, CTR, GCM, CCM, XTS, OMAC1, and CMAC. Ciphertext and plaintext data are transferred through the AHB Host port.

43.4.2 Hash

The Hash accelerator supports the SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 hashes. HMAC processing is supported for all hashes. Message data is read through the AHB Host port.

43.4.3 Public Key

The Public Key (PK) accelerator implements the following algorithms: RSA, DSA, DH, ECDH, ECDSA, EdDSA, J-PAKE and SRP. RSA, DSA and DH algorithms support up to 4096 bit key sizes. Standard ECC curves include: P-256, P-384 and 25519. Other ECC curves are supported through custom parameters including Brainpool, Koblitz, Montgomery, and Edwards.

Up to 16 Kb of system RAM is used for PK inputs, outputs, and scratch pad RAM. The location and size of the PK RAM area is programmable. Likewise 8 Kb of Boot ROM is used for PK microcode. The AHB Host port is used for fetching.

43.5 TRNG

Cryptographically secure random number generation is provided by a NIST 800-90B compliant True Random Number Generator (TRNG) also known as Non-Deterministic Random Number Generator (NDRNG). The TRNG is used to seed a NIST 800-90A compliant Deterministic Random Bit Generator (DRBG).

43.6 PUF Interface

A dedicated key input is included for interfacing with a Physically Unclonable Function (PUF). The PUF key can be selected for local storage encryption and is never visible to the CPU or the debugger.

43.7 Peripheral Dependencies

Table 43-1. Peripheral Dependencies

Peripheral Name	Base Address	NVIC IRQ Index:Source	MCLK AHB/APB Clock Enable Bus:Register:Bit	GCLK Peripheral Channel Control Register	PAC Peripheral Identifier PERID:Register:Bit	DMA Trigger Index:Source	Event System Type:Event: Register:Path	Power Domain
HSM	0x4F000000	139 : CRYPTOMASTER/RNG/PUBKEY	AHB: CLKMSK[1] MASK30	---	48 STATUS1 PERID16			PD_CORE_SW

43.8 Register Summary

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x00	CTRLA	31:24								
		23:16								
		15:8								
		7:0		RUNSTDBY				PRIV	ENABLE	SWRST
0x04 ... 0x37	Reserved									
0x38	STATUSA	31:24								
		23:16								
		15:8								
		7:0								OTPLOCK
0x3C ... 0x5B	Reserved									
0x5C	SYNCBUSY	31:24								
		23:16								
		15:8								
		7:0							ENABLE	SWRST
0x60 ... 0xDF	Reserved									
0xE0	WPCTRL	31:24	WPKEY[23:16]							
		23:16	WPKEY[15:8]							
		15:8	WPKEY[7:0]							
		7:0							WPLCK	WPEN
0xE4 ... 0xFFFF	Reserved									
0x010000	CryptoMaster_DMA_Registers_FETCH_ADDR_LSB	31:24	FETCH_ADDR_LSB[31:24]							
		23:16	FETCH_ADDR_LSB[23:16]							
		15:8	FETCH_ADDR_LSB[15:8]							
		7:0	FETCH_ADDR_LSB[7:0]							
0x010004	CryptoMaster_DMA_Registers_FETCH_ADDR_MSB	31:24	FETCH_ADDR_MSB[31:24]							
		23:16	FETCH_ADDR_MSB[23:16]							
		15:8	FETCH_ADDR_MSB[15:8]							
		7:0	FETCH_ADDR_MSB[7:0]							
0x010008	CryptoMaster_DMA_Registers_FETCH_LEN	31:24		FETCH_ZPAD DING	FETCH_REALI GN	FETCH_CSTAD DR	FETCH_LEN[27:24]			
		23:16	FETCH_LEN[23:16]							
		15:8	FETCH_LEN[15:8]							
		7:0	FETCH_LEN[7:0]							
0x01000C	CryptoMaster_DMA_Registers_FETCH_TAG	31:24	FETCH_TAG[31:24]							
		23:16	FETCH_TAG[23:16]							
		15:8	FETCH_TAG[15:8]							
		7:0	FETCH_TAG[7:0]							
0x010010	CryptoMaster_DMA_Registers_PUSH_ADDR_LSB	31:24	PUSH_ADDR_LSB[31:24]							
		23:16	PUSH_ADDR_LSB[23:16]							
		15:8	PUSH_ADDR_LSB[15:8]							
		7:0	PUSH_ADDR_LSB[7:0]							
0x010014	CryptoMaster_DMA_Registers_PUSH_ADDR_MSB	31:24	PUSH_ADDR_MSB[31:24]							
		23:16	PUSH_ADDR_MSB[23:16]							
		15:8	PUSH_ADDR_MSB[15:8]							
		7:0	PUSH_ADDR_MSB[7:0]							
0x010018	CryptoMaster_DMA_Registers_PUSH_LEN	31:24		PUSH_DISCAR D	PUSH_REALIG N	PUSH_CSTAD DR	PUSH_LEN[27:24]			
		23:16	PUSH_LEN[23:16]							
		15:8	PUSH_LEN[15:8]							
		7:0	PUSH_LEN[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01001C	CryptoMaster_DMA_Registers_INT_EN	31:24								
		23:16								
		15:8								
		7:0			INT_EN[5:0]					
0x010020	CryptoMaster_DMA_Registers_INT_EN_SET	31:24								
		23:16								
		15:8								
		7:0			INT_ENSET[5:0]					
0x010024	CryptoMaster_DMA_Registers_INT_EN_CLR	31:24								
		23:16								
		15:8								
		7:0			INT_ENCLR[5:0]					
0x010028	CryptoMaster_DMA_Registers_INT_STA_TRAW	31:24								
		23:16								
		15:8								
		7:0			INT_STA_TRAW[5:0]					
0x01002C	CryptoMaster_DMA_Registers_INT_STA_T	31:24								
		23:16								
		15:8								
		7:0			INT_STA_T[5:0]					
0x010030	CryptoMaster_DMA_Registers_INT_STA_TCLR	31:24								
		23:16								
		15:8								
		7:0			INT_STA_TCLR[5:0]					
0x010034	CryptoMaster_DMA_Registers_CONFIG	31:24								
		23:16								
		15:8								
		7:0				SOFT_RST	PUSH_STOP	FETCH_STOP	PUSH_CTRL_INDIRECT	FETCH_CTRL_INDIRECT
0x010038	CryptoMaster_DMA_Registers_START	31:24								
		23:16								
		15:8								
		7:0							START_PUSH	START_FETCH
0x01003C	CryptoMaster_DMA_Registers_STATUS	31:24	PUSH_NBDATA[15:8]							
		23:16	PUSH_NBDATA[7:0]							
		15:8								
		7:0		SOFT_RST_BUSY	PUSH_WAITING_FIFO	FETCH_NOT_EMPTY			PUSH_BUSY	FETCH_BUSY
0x010040 ... 0x0103FF	Reserved									
0x010400	CryptoMaster_HWConf_Registers_INC_LIPS_HW_CFG	31:24								
		23:16								
		15:8			BA422_KASUMI_INCLUDED	BA423_SNOW3G_INCLUDED	BA420_HP_CHACHA_POLY_INCLUDED	BA431_NDRNG_INCLUDED	BA414EP_PKE_INCLUDED	BA419_SM4_INCLUDED
		7:0	BA421_ZUC_INCLUDED	BA418_SHA3_INCLUDED	BA417_CHACHA_POLY_INCLUDED	BA413_HASH_INCLUDED	BA412_DES_INCLUDED	BA416_HP_AES_S_XTS_INCLUDED	BA415_HP_AES_S_GCM_INCLUDED	BA411_AES_INCLUDED
0x010404	CryptoMaster_HWConf_Registers_BA411E_AES_HW_CFG_1	31:24				BA411E_AES_HW_CFG_GLITCH_PROTECTION	BA411E_AES_HW_CFG_CSWITCH	BA411E_AES_HW_CFG_KEYSIZE[2:0]		
		23:16							BA411E_AES_HW_CFG_MASKING	BA411E_AES_HW_CFG_CS
		15:8								BA411E_AES_HW_CFG_MODE[8]
		7:0	BA411E_AES_HW_CFG_MODE[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x010408	CryptoMaster_HWConfg_Registers_BA411E_AES_HW_CFG_2	31:24	BA411E_AES_HW_CFG_2_IKGKEYS[3:0]							
		23:16	BA411E_AES_HW_CFG_2_EXTKEYS[3:0]							
		15:8	BA411E_AES_HW_CFG_2_CTRSIZE[15:8]							
		7:0	BA411E_AES_HW_CFG_2_CTRSIZE[7:0]							
0x01040C	CryptoMaster_HWConfg_Registers_BA413_HASH_HW_CFG	31:24	BA413_HASH_HW_CFG_IKGKEYS[3:0]							
		23:16	BA413_HASH_HW_CFG_EXTKEYS[3:0]							
		15:8	BA413_HASH_HW_CFG_MASK[6:0]							
		7:0	BA413_HASH_HW_CFG_MASK[6:0]							
0x010410	CryptoMaster_HWConfg_Registers_BA418_SHA3_HW_CFG	31:24								
		23:16								
		15:8								
		7:0								
0x010414	CryptoMaster_HWConfg_Registers_BA419_SM4_HW_CFG	31:24								
		23:16								
		15:8								
		7:0								
0x010418	CryptoMaster_HWConfg_Registers_BA424_ARIA_HW_CFG	31:24								
		23:16								
		15:8								
		7:0								
0x01041C ... 0x010FFF	Reserved									
0x011000	RNG_Control_Registers_Control	31:24								
		23:16								
		15:8	AIS31TestSel	HealthTestSel	AIS31Bypass	HealthTestBy pass	ForceActiveR Os	IntEnAlm	IntEnPre	SoftRst
		7:0	IntEnFull		IntEnProp	IntEnRep	CondBypass	TestEn	LFSREn	Enable
0x011004	RNG_Control_Registers_FIFOLevel	31:24								
		23:16								
		15:8								
		7:0								
0x011008	RNG_Control_Registers_FIFOThreshold	31:24								
		23:16								
		15:8								
		7:0								
0x01100C	RNG_Control_Registers_FIFODepth	31:24								
		23:16								
		15:8								
		7:0								
0x011010	RNG_Control_Registers_Key0	31:24								
		23:16								
		15:8								
		7:0								
0x011014	RNG_Control_Registers_Key1	31:24								
		23:16								
		15:8								
		7:0								
0x011018	RNG_Control_Registers_Key2	31:24								
		23:16								
		15:8								
		7:0								

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x01101C	RNG_Control_Registers_Key3	31:24	Key3[31:24]							
		23:16	Key3[23:16]							
		15:8	Key3[15:8]							
		7:0	Key3[7:0]							
0x011020	RNG_Control_Registers_TestData	31:24	TestData[31:24]							
		23:16	TestData[23:16]							
		15:8	TestData[15:8]							
		7:0	TestData[7:0]							
0x011024	RNG_Control_Registers_RepeatThreshold	31:24								
		23:16								
		15:8								
		7:0	RepeatThreshold[5:0]							
0x011028	RNG_Control_Registers_PropThreshold	31:24								
		23:16								
		15:8							PropThreshold[9:8]	
		7:0	PropThreshold[7:0]							
0x01102C ...	Reserved									
0x01102F										
0x011030	RNG_Control_Registers_Status	31:24								
		23:16								
		15:8					FifoAccFail	StartUpFail	AlmInt	PreInt
		7:0	FullInt		PropFail	RepFail	State[2:0]		TestDataBusy	
0x011034	RNG_Control_Registers_InitWaitVal	31:24								
		23:16								
		15:8	InitWaitVal[15:8]							
		7:0	InitWaitVal[7:0]							
0x011038	RNG_Control_Registers_DisableOsc0	31:24	DisableOsc0[31:24]							
		23:16	DisableOsc0[23:16]							
		15:8	DisableOsc0[15:8]							
		7:0	DisableOsc0[7:0]							
0x01103C	RNG_Control_Registers_DisableOsc1	31:24	DisableOsc1[31:24]							
		23:16	DisableOsc1[23:16]							
		15:8	DisableOsc1[15:8]							
		7:0	DisableOsc1[7:0]							
0x011040	RNG_Control_Registers_SwOffTmrVal	31:24								
		23:16								
		15:8	SwOffTmrVal[15:8]							
		7:0	SwOffTmrVal[7:0]							
0x011044	RNG_Control_Registers_ClkDiv	31:24								
		23:16								
		15:8								
		7:0	ClkDiv[7:0]							
0x011048	RNG_Control_Registers_AIS31Conf0	31:24	OnlineThreshold[15:8]							
		23:16	OnlineThreshold[7:0]							
		15:8	StartUpThreshold[15:8]							
		7:0	StartUpThreshold[7:0]							
0x01104C	RNG_Control_Registers_AIS31Conf1	31:24	HExpectedValue[15:8]							
		23:16	HExpectedValue[7:0]							
		15:8	OnlineRepThreshold[15:8]							
		7:0	OnlineRepThreshold[7:0]							
0x011050	RNG_Control_Registers_AIS31Conf2	31:24	HMax[15:8]							
		23:16	HMax[7:0]							
		15:8	HMin[15:8]							
		7:0	HMin[7:0]							
0x011054	RNG_Control_Registers_AIS31Status	31:24								
		23:16							PrelimNoiseAl armRep	PrelimNoiseAl armRng
		15:8	NumPrelimAlarms[15:8]							
		7:0	NumPrelimAlarms[7:0]							

Register Summary (continued)

Offset	Name	Bit Pos.	7	6	5	4	3	2	1	0
0x011058	RNG_Control_Registers_HwConfig	31:24								
		23:16								
		15:8							AIS31Full	AIS31
		7:0	NumbOfRings[7:0]							
0x01105C ... 0x011FFF	Reserved									
0x012000	PK_Registers_Pointers	31:24					OpPtrN[3:0]			
		23:16					OpPtrC[3:0]			
		15:8					OpPtrB[3:0]			
		7:0					OpPtrA[3:0]			
0x012004	PK_Registers_Command	31:24	CalcR2	FlagB	FlagA	SwapBytes		Edwards	RandProj	RandKE
		23:16		SelCurve[2:0]			RandMod		OpBytesM1[9:8]	
		15:8	OpBytesM1[7:0]							
		7:0	FieldF	OpeAddr[6:0]						
0x012008	PK_Registers_Control	31:24								
		23:16								
		15:8								
		7:0							ClearIRQ	Start
0x01200C	PK_Registers_Status	31:24				FailPtr[4:0]				
		23:16							IntrptStatus	PK_Busy
		15:8	ErrorFlags[11:4]							
		7:0	ErrorFlags[3:0]							
0x012010 ... 0x012013	Reserved									
0x012014	PK_Registers_Timer	31:24	Timer[30:23]							
		23:16	Timer[22:15]							
		15:8	Timer[14:7]							
		7:0	Timer[6:0]							
0x012018	PK_Registers_HwConfig	31:24	DisableCM	DisableClrMem	DisableSMx				AHBMaster	X25519
		23:16	P192	P521	P384	P256		ECC	BinaryField	PrimeField
		15:8	NbMult[3:0]				MaxOpSize[11:8]			
		7:0	MaxOpSize[7:0]							
0x01201C	PK_Registers_OpSize	31:24								
		23:16								
		15:8				OpSize[12:8]				
		7:0	OpSize[7:0]							
0x012020	PK_Registers_MemOffset	31:24	MemOffset[31:24]							
		23:16	MemOffset[23:16]							
		15:8	MemOffset[15:8]							
		7:0	MemOffset[7:0]							
0x012024	PK_Registers_MicroCodeOffset	31:24	MicroCodeOffset[31:24]							
		23:16	MicroCodeOffset[23:16]							
		15:8	MicroCodeOffset[15:8]							
		7:0	MicroCodeOffset[7:0]							

43.8.1 Control Register A

Name: CTRLA
Offset: 0x0
Reset: 0x00000000
Property: R/W

Table 43-2. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
		RUNSTDBY				PRIV	ENABLE	SWRST
Access		R/W				R/W	R/W	R/S/HC
Reset		0				0	0	0

Bit 6 – RUNSTDBY Run In Standby bit

Value	Name	Description
1		If enabled, the CAM clock remains requested in Standby Sleep mode.
0		CAM clock is disabled in Standby Sleep mode, clock requests are de-asserted after any pending bus transactions or requests are completed.

Bit 2 – PRIV Privileged Access Only - This bit affects all Wrapper and EIP registers. This bit affects all Wrapper and EIP registers.

Value	Name	Description
1		CAM registers only accessible in privileged mode. Master Requests are also privileged.
0		CAM registers accessible in privileged and unprivileged modes. Master Requests are NOT privileged

Bit 1 – ENABLE Setting this bit also sets the SYNCBUSY.ENABLE to 1. SYNCBUSY.ENABLE stays 1 until Enable / Disable sequence completes.

Setting this bit also sets the SYNCBUSY.ENABLE to 1. SYNCBUSY.ENABLE stays 1 until Enable or Disable sequence completes.

Value	Name	Description
1		CAM clock is requested and macro is enabled.
0		CAM clock is not requested and macros is disabled.

Bit 0 – SWRST Write '1' to reset registers and internal state. Writing '0' has no effect. SWRST stays high until reset completes. Setting this bit also sets the SYNCBUSY.SWRST to 1. SYNCBUSY.SWRST stays 1 until reset sequence completes.

Write '1' to reset registers and internal state. Writing '0' has no effect. SWRST stays high until reset completes. Setting this bit also sets the SYNCBUSY.SWRST to 1. SYNCBUSY.SWRST stays 1 until reset sequence completes.

Value	Description
0x0	No effect
0x1	Reset registers and internal state

43.8.2 STATUS Register A

Name: STATUSA
Offset: 0x38
Reset: 0x00000000
Property: R

Table 43-3. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								OTPLOCK
Access								R/HS/HC
Reset								0

Bit 0 – OTPLOCK OTP Lock bit

Value	Name	Description
1		OTP lock bit set. Device not in IC manufacturing.

43.8.3 SYNCBUSY Register

Name: SYNCBUSY
Offset: 0x5c
Reset: 0x00000000
Property: R

Table 43-4. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							ENABLE	SWRST
Access							R/HS/HC	R/HS/HC
Reset							0	0

Bit 1 – ENABLE Macro Enable Synchronization Busy

Value	Name	Description
1		ENABLE synchronization is busy.
0		ENABLE synchronization is NOT busy.

Bit 0 – SWRST Software Reset Synchronization Busy1 = SWRST synchronization is busy0 = SWRST synchronization is NOT busy

Value	Description
0x0	SWRST synchronization is not busy
0x1	SWRST synchronization is busy

43.8.4 WRITE PROTECTION CONTROL REGISTER

Name: WPCTRL
Offset: 0xe0
Reset: 0x00000000
Property: R/W

Table 43-5. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
							WPLCK	WPEN
Access							R/W	R/W
Reset							0	0

Bits 31:8 – WPKEY[23:0] Writing a value other than CAM_WPCTRL_KEY to this field cancels the write operation to this register and generates a slave bus error. This field always returns 0 on read.

Writing a value other than 0x43414D to this field cancels the write operation to this register and generates a client bus error. This field always returns 0 on read.

Bit 1 – WPLCK WPCTRL Write Lock Bit

Value	Description
0x0	WPCTRL is not write protected.
0x1	The WPCTRL register is write protected and any non-debugger writes to this register generates a client bus error. When set, the debugger is not allowed to clear this bit and the access generates a client bus error. This bit can be cleared only by a hardware reset, the SWRST has no effect.

Value	Name	Description
1		WPCTRL register is write protected and any non-debugger writes to this register generates a slave bus error. When set, the debugger is not allowed to clear this bit and the access generates a slave bus error. This bit can be cleared only by a hardware reset, the SWRST has no effect.
0		WPCTRL is not write protected.

Bit 0 – WPEN Write Protection Enable

Value	Description
0x0	Register write protection disabled.
0x1	Register write protection enabled. Non-debugger writes to registers marked with PAC write protection property are canceled and generate a client bus error.

Value	Name	Description
1		Register write protection enabled. Non-debugger writes to registers marked with PAC write protection property are canceled and generate a slave bus error.
0		Register write protection disabled.

43.8.5 [CryptoMaster_DMA_Registers__FETCH_ADDR_LSB]

Name: CryptoMaster_DMA_Registers__FETCH_ADDR_LSB
Offset: 0x10000
Reset: 0x00000000
Property: RW

Table 43-6. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FETCH_ADDR_LSB[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FETCH_ADDR_LSB[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FETCH_ADDR_LSB[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FETCH_ADDR_LSB[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FETCH_ADDR_LSB[31:0]

43.8.6 [CryptoMaster_DMA_Registers__FETCH_ADDR_MSB]

Name: CryptoMaster_DMA_Registers__FETCH_ADDR_MSB
Offset: 0x10004
Reset: 0x00000000
Property: RW

Table 43-7. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FETCH_ADDR_MSB[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FETCH_ADDR_MSB[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FETCH_ADDR_MSB[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FETCH_ADDR_MSB[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FETCH_ADDR_MSB[31:0]

43.8.7 [CryptoMaster_DMA_Registers__FETCH_LEN]

Name: CryptoMaster_DMA_Registers__FETCH_LEN
Offset: 0x10008
Reset: 0x00000000
Property: RW

Table 43-8. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		FETCH_ZPAD DING	FETCH_REALI GN	FETCH_CSTA DDR	FETCH_LEN[27:24]			
Access		W	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FETCH_LEN[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FETCH_LEN[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FETCH_LEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 30 – FETCH_ZPADDING

Bit 29 – FETCH_REALIGN

Bit 28 – FETCH_CSTADDR

Bits 27:0 – FETCH_LEN[27:0]

43.8.8 [CryptoMaster_DMA_Registers__FETCH_TAG]

Name: CryptoMaster_DMA_Registers__FETCH_TAG
Offset: 0x1000C
Reset: 0x00000000
Property: RW

Table 43-9. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FETCH_TAG[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FETCH_TAG[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FETCH_TAG[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FETCH_TAG[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FETCH_TAG[31:0]

43.8.9 [CryptoMaster_DMA_Registers__PUSH_ADDR_LSB]

Name: CryptoMaster_DMA_Registers__PUSH_ADDR_LSB
Offset: 0x10010
Reset: 0x00000000
Property: RW

Table 43-10. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PUSH_ADDR_LSB[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PUSH_ADDR_LSB[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PUSH_ADDR_LSB[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PUSH_ADDR_LSB[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – PUSH_ADDR_LSB[31:0]

43.8.10 [CryptoMaster_DMA_Registers__PUSH_ADDR_MSB]

Name: CryptoMaster_DMA_Registers__PUSH_ADDR_MSB
Offset: 0x10014
Reset: 0x00000000
Property: RW

Table 43-11. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PUSH_ADDR_MSB[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PUSH_ADDR_MSB[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PUSH_ADDR_MSB[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PUSH_ADDR_MSB[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 - PUSH_ADDR_MSB[31:0]

43.8.11 [CryptoMaster_DMA_Registers__PUSH_LEN]

Name: CryptoMaster_DMA_Registers__PUSH_LEN
Offset: 0x10018
Reset: 0x00000000
Property: RW

Table 43-12. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
		PUSH_DISCARD	PUSH_REALIGN	PUSH_CSTADDR	PUSH_LEN[27:24]			
Access		RW	RW	RW	RW	RW	RW	RW
Reset		0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PUSH_LEN[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	PUSH_LEN[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	PUSH_LEN[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 30 – PUSH_DISCARD

Bit 29 – PUSH_REALIGN

Bit 28 – PUSH_CSTADDR

Bits 27:0 – PUSH_LEN[27:0]

43.8.12 [CryptoMaster_DMA_Registers__INT_EN]

Name: CryptoMaster_DMA_Registers__INT_EN
Offset: 0x1001C
Reset: 0x00000000
Property: RW

Table 43-13. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			INT_EN[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			0	0	0	0	0	0

Bits 5:0 – INT_EN[5:0]

43.8.13 [CryptoMaster_DMA_Registers__INT_ENSET]

Name: CryptoMaster_DMA_Registers__INT_ENSET
Offset: 0x10020
Reset: 0x00000000
Property: W

Table 43-14. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			INT_ENSET[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	0

Bits 5:0 – INT_ENSET[5:0]

43.8.14 [CryptoMaster_DMA_Registers__INT_ENCLR]

Name: CryptoMaster_DMA_Registers__INT_ENCLR
Offset: 0x10024
Reset: 0x00000000
Property: W

Table 43-15. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			INT_ENCLR[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	0

Bits 5:0 – INT_ENCLR[5:0]

43.8.15 [CryptoMaster_DMA_Registers__INT_STATRAW]

Name: CryptoMaster_DMA_Registers__INT_STATRAW
Offset: 0x10028
Reset: 0x00000000
Property: R

Table 43-16. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			INT_STATRAW[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 5:0 – INT_STATRAW[5:0]

43.8.16 [CryptoMaster_DMA_Registers__INT_STAT]

Name: CryptoMaster_DMA_Registers__INT_STAT
Offset: 0x1002C
Reset: 0x00000000
Property: R

Table 43-17. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			INT_STAT[5:0]					
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bits 5:0 – INT_STAT[5:0]

43.8.17 [CryptoMaster_DMA_Registers__INT_STATCLR]

Name: CryptoMaster_DMA_Registers__INT_STATCLR
Offset: 0x10030
Reset: 0x00000000
Property: W

Table 43-18. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			INT_STATCLR[5:0]					
Access			W	W	W	W	W	W
Reset			0	0	0	0	0	0

Bits 5:0 – INT_STATCLR[5:0]

43.8.18 [CryptoMaster_DMA_Registers__CONFIG]

Name: CryptoMaster_DMA_Registers__CONFIG
Offset: 0x10034
Reset: 0x00000000
Property: RW

Table 43-19. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
				SOFT_RST	PUSH_STOP	FETCH_STOP	PUSH_CTRL_I NDIRECT	FETCH_CTRL_ INDIRECT
Access				RW	RW	RW	RW	RW
Reset				0	0	0	0	0

Bit 4 – SOFT_RST

Bit 3 – PUSH_STOP

Bit 2 – FETCH_STOP

Bit 1 – PUSH_CTRL_INDIRECT

Bit 0 – FETCH_CTRL_INDIRECT

43.8.19 [CryptoMaster_DMA_Registers__START]

Name: CryptoMaster_DMA_Registers__START
Offset: 0x10038
Reset: 0x00000000
Property: W

Table 43-20. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							START_PUSH	START_FETCH
Access							W	W
Reset							0	0

Bit 1 – START_PUSH

Bit 0 – START_FETCH

43.8.20 [CryptoMaster_DMA_Registers__STATUS]

Name: CryptoMaster_DMA_Registers__STATUS
Offset: 0x1003C
Reset: 0x00000000
Property: R

Table 43-21. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	PUSH_NBDATA[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	PUSH_NBDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
		SOFT_RST_BU SY	PUSH_WAITI NGFIFO	FETCH_NOT_ EMPTY			PUSH_BUSY	FETCH_BUSY
Access		R	R	R			R	R
Reset		0	0	0			0	0

Bits 31:16 – PUSH_NBDATA[15:0]

Bit 6 – SOFT_RST_BUSY

Bit 5 – PUSH_WAITINGFIFO

Bit 4 – FETCH_NOT_EMPTY

Bit 1 – PUSH_BUSY

Bit 0 – FETCH_BUSY

43.8.21 [CryptoMaster_HWConf_Registers__INCL_IPS_HW_CFG]

Name: CryptoMaster_HWConf_Registers__INCL_IPS_HW_CFG
Offset: 0x10400
Reset: 0x00000611
Property: R

Table 43-22. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
			BA422_KASU MI_INCLUDED	BA423_SNO W3G_INCLUDED	BA420_HP_C HACHAPOLY_ INCLUDED	BA431_NDRN G_INCLUDED	BA414EP_PKE _INCLUDED	BA419_SM4_I NCLUDED
Access			R	R	R	R	R	R
Reset			0	0	0	1	1	0
Bit	7	6	5	4	3	2	1	0
	BA421_ZUC_I NCLUDED	BA418_SHA3 INCLUDED	BA417_CHAC HAPOLY_INC LUDED	BA413_HASH _INCLUDED	BA412_DES_I NCLUDED	BA416_HP_AE S_XTS_INCLU DED	BA415_HP_AE S_GCM_INCL UDED	BA411_AES_I NCLUDED
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	1	0	0	0	1

Bit 13 – BA422_KASUMI_INCLUDED Generic g_IncludeKasumi value. BA422-Kasumi IP included if set

Bit 12 – BA423_SNOW3G_INCLUDED Generic g_IncludeSnow3G value. BA423-Snow3G IP included if set

Bit 11 – BA420_HP_CHACHAPOLY_INCLUDED Generic g_IncludeHPChaChaPoly value. BA420-HP-ChaChaPoly IP included if set

Bit 10 – BA431_NDRNG_INCLUDED Generic g_IncludeNDRNG value. BA431-NDRNG IP included if set

Bit 9 – BA414EP_PKE_INCLUDED Generic g_IncludePKE value. BA414EP-PKE IP included if set

Bit 8 – BA419_SM4_INCLUDED Generic g_IncludeSM4 value. BA419-SM4 IP included if set

Bit 7 – BA421_ZUC_INCLUDED Generic g_IncludeZUC value. BA421-ZUC IP included if set

Bit 6 – BA418_SHA3_INCLUDED Generic g_IncludeSHA3 value. BA418-SHA3 IP included if set

Bit 5 – BA417_CHACHAPOLY_INCLUDED Generic g_IncludeChachaPoly value. BA417-ChaChaPoly IP included if set

Bit 4 – BA413_HASH_INCLUDED Generic g_IncludeHASH value. BA413-HASH IP included if set

Bit 3 – BA412_DES_INCLUDED Generic g_IncludeDES value. BA412-3DES IP included if set

Bit 2 – BA416_HP_AES_XTS_INCLUDED Generic g_IncludeAESXTS value. BA416-HP-AES-XTS IP included if set

Bit 1 – BA415_HP_AES_GCM_INCLUDED Generic g_IncludeAESGCM value. BA415-HP-AES-GCM IP included if set

Bit 0 – BA411_AES_INCLUDED Generic g_IncludeAES value. BA411E-AES IP included if set

43.8.22 [CryptoMaster_HWConf_Registers__BA411E_AES_HW_CFG_1]

Name: CryptoMaster_HWConf_Registers__BA411E_AES_HW_CFG_1
Offset: 0x10404
Reset: 0x1F0301FF
Property: R

Table 43-23. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				BA411E_AES_HW_CFG_GLITCHPROTECTION	BA411E_AES_HW_CFG_CXSWITCH	BA411E_AES_HW_CFG_KEYSIZE[2:0]		
Access				R	R	R	R	R
Reset				1	1	1	1	1
Bit	23	22	21	20	19	18	17	16
							BA411E_AES_HW_CFG_MASKING	BA411E_AES_HW_CFG_CS
Access							R	R
Reset							1	1
Bit	15	14	13	12	11	10	9	8
								BA411E_AES_HW_CFG_MODE[8]
Access								R
Reset								1
Bit	7	6	5	4	3	2	1	0
	BA411E_AES_HW_CFG_MODE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bit 28 – BA411E_AES_HW_CFG_GLITCHPROTECTION Generic g_GlitchProtection value. BA411E-AES engine configuration.

Bit 27 – BA411E_AES_HW_CFG_CXSWITCH Generic g_CxSwitch value. BA411E-AES engine configuration.

Bits 26:24 – BA411E_AES_HW_CFG_KEYSIZE[2:0] Generic g_KeySize value. BA411E-AES engine configuration.

Bit 17 – BA411E_AES_HW_CFG_MASKING Generic g_UseMasking value. BA411E-AES engine configuration.

Bit 16 – BA411E_AES_HW_CFG_CS Generic g_CS value. BA411E-AES engine configuration.

Bits 8:0 – BA411E_AES_HW_CFG_MODE[8:0] Generic g_AesModesPoss value. BA411E-AES engine configuration.

43.8.23 Generic g_CtrSize value. BA411E-AES engine configuration.

Name: CryptoMaster_HWConf_Registers_BA411E_AES_HW_CFG_2
Offset: 0x10408
Reset: 0x00200080
Property: R

Table 43-24. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					BA411E_AES_HW_CFG_2_IKGKEYS[3:0]			
Access					R	R	R	R
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BA411E_AES_HW_CFG_2_EXTKEYS[3:0]							
Access	R	R	R	R				
Reset	0	0	1	0				
Bit	15	14	13	12	11	10	9	8
	BA411E_AES_HW_CFG_2_CTRSIZE[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	BA411E_AES_HW_CFG_2_CTRSIZE[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	0	0	0	0	0	0	0

Bits 27:24 – BA411E_AES_HW_CFG_2_IKGKEYS[3:0] Generic g_IKG_nb_AES_keys value. BA411E-AES engine configuration.

Bits 23:20 – BA411E_AES_HW_CFG_2_EXTKEYS[3:0] Generic g_Ext_nb_AES_keys value. BA411E-AES engine configuration.

Bits 15:0 – BA411E_AES_HW_CFG_2_CTRSIZE[15:0] Generic g_CtrSize value. BA411E-AES engine configuration.

43.8.24 [CryptoMaster_HWConf_Registers__BA413_HASH_HW_CFG]

Name: CryptoMaster_HWConf_Registers__BA413_HASH_HW_CFG
Offset: 0x1040C
Reset: 0x0013003E
Property: R

Table 43-25. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
					BA413_HASH_HW_CFG_IKGKEYS[3:0]			
Access					R	R	R	R
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
	BA413_HASH_HW_CFG_EXTKEYS[3:0]					BA413_HASH_HW_CFG_VERIFYDIGEST	BA413_HASH_HW_CFG_HMAC	BA413_HASH_HW_CFG_PADDING
Access	R	R	R	R		R	R	R
Reset	0	0	0	1		0	1	1
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
		BA413_HASH_HW_CFG_MASK[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	1	1	1	1	1	0

Bits 27:24 – BA413_HASH_HW_CFG_IKGKEYS[3:0] Generic g_IKG_nb_Hash_keys value. BA413-Hash engine configuration.

Bits 23:20 – BA413_HASH_HW_CFG_EXTKEYS[3:0] Generic g_Ext_nb_Hash_keys value. BA413-Hash engine configuration.

Bit 18 – BA413_HASH_HW_CFG_VERIFYDIGEST Generic g_HashVerifyDigest value. BA413-Hash engine configuration.

Bit 17 – BA413_HASH_HW_CFG_HMAC Generic g_HMAC_enabled value. BA413-Hash engine configuration.

Bit 16 – BA413_HASH_HW_CFG_PADDING Generic g_HashPadding value. BA413-Hash engine configuration.

Bits 6:0 – BA413_HASH_HW_CFG_MASK[6:0] Generic g_HashMaskFunc value. BA413-Hash engine configuration.

43.8.25 Generic g_Sha3CtxtEn value. BA418-SHA3 configuration.

Name: CryptoMaster_HWConf_Registers_BA418_SHA3_HW_CFG
Offset: 0x10410
Reset: 0x00000000
Property: R

Table 43-26. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
								BA418_SHA3_ HW_CFG
Access								R
Reset								0

Bit 0 – BA418_SHA3_HW_CFG Generic g_Sha3CtxtEn value. BA418-SHA3 configuration.

43.8.26 Generic g_SM4ModesPoss value. BA419-SM4 engine configuration.

Name: CryptoMaster_HWConf_Registers_BA419_SM4_HW_CFG
Offset: 0x10414
Reset: 0x000001FF
Property: R

Table 43-27. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
								BA419_SM4_HW_CFG[8]
Access								R
Reset								1
Bit	7	6	5	4	3	2	1	0
	BA419_SM4_HW_CFG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	1	1	1	1	1	1	1	1

Bits 8:0 - BA419_SM4_HW_CFG[8:0] Generic g_SM4ModesPoss value. BA419-SM4 engine configuration.

43.8.27 Generic g_aria_modePoss value. BA424-Aria engine configuration.

Name: CryptoMaster_HWConf_Registers_BA424_ARIA_HW_CFG
Offset: 0x10418
Reset: 0x0000017F
Property: R

Table 43-28. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
								BA424_ARIA_HW_CFG[8]
Access								R
Reset								1
Bit	7	6	5	4	3	2	1	0
	BA424_ARIA_HW_CFG[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	1	1	1	1	1	1	1

Bits 8:0 - BA424_ARIA_HW_CFG[8:0] Generic g_aria_modePoss value. BA424-Aria engine configuration.

43.8.28 Control register

Name: RNG_Control_Registers_Control
Offset: 0x11000
Reset: 0x00040000
Property: RW

Table 43-29. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access				FifoWriteStartUp		Nb128BitBlocks[3:0]		
Reset				RW	RW	RW	RW	RW
				0	0	1	0	0
Bit	15	14	13	12	11	10	9	8
Access	AIS31TestSel	HealthTestSel	AIS31Bypass	HealthTestBypass	ForceActiveROs	IntEnAlm	IntEnPre	SoftRst
Reset	RW	RW	RW	RW	RW	RW	RW	RW
	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Access	IntEnFull		IntEnProp	IntEnRep	CondBypass	TestEn	LFSREn	Enable
Reset	RW		RW	RW	RW	RW	RW	RW
	0		0	0	0	0	0	0

Bit 20 – FifoWriteStartUp Enable write of the samples in the FIFO during start-up.

Bits 19:16 – Nb128BitBlocks[3:0] Number of 128 bit blocks used in AES-CBCMAC post-processing. This value cannot be zero.

Bit 15 – AIS31TestSel Select input to the AIS31 test module: 0: Before conditioning, 1: After conditioning.

Bit 14 – HealthTestSel Select input to health test module: 0: Before conditioning, 1: After conditioning.

Bit 13 – AIS31Bypass Bypass AIS31 tests such that the results of the start-up and online tests do not affect the FSM state.

Bit 12 – HealthTestBypass Bypass NIST tests such that the results of the start-up and online test do not affect the FSM state.

Bit 11 – ForceActiveROs Force oscillators to run when FIFO is full.

Bit 10 – IntEnAlm Interrupt enable for AIS31 noise alarm.

Bit 9 – IntEnPre Interrupt enable for AIS31 preliminary noise alarm.

Bit 8 – SoftRst Software reset: 0: Normal mode, 1: The continuous test, the conditioning function and the FIFO are reset. This bit is not cleared automatically.

Bit 7 – IntEnFull Interrupt enable for FIFO full.

Bit 5 – IntEnProp Interrupt enable for Adaptive Proportion Test failure (1024-sample window).

Bit 4 – IntEnRep Interrupt enable for Repetition Count Test failure.

Bit 3 – CondBypass 0: the conditioning function is used (normal mode), 1: the conditioning function is bypassed (to observe entropy source directly).

Bit 2 – TestEn Select input for conditioning function and continuous tests: 0: Noise source (normal mode), 1: Test data register (test mode).

Bit 1 – LFSREn Select between the NDRNG with asynchronous free running oscillators (when '0') and the Pseudo-Random generator with synchronous oscillators for simulation purpose (when '1').

Bit 0 – Enable Enable the NDRNG.

43.8.29 FIFO level register.

Name: RNG_Control_Registers_FIFOLevel
Offset: 0x11004
Reset: 0x00000000
Property: RW

Table 43-30. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FIFOLevel[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FIFOLevel[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FIFOLevel[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FIFOLevel[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – FIFOLevel[31:0] Number of 32 bits words of random available in the FIFO. Any write to this register clears the FullInt flag in the Status register.

43.8.30 FIFO threshold register.

Name: RNG_Control_Registers_FIFOThreshold
Offset: 0x11008
Reset: 0x00000001
Property: RW

Table 43-31. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							FIFOThreshold[1:0]	
Access							RW	RW
Reset							0	1

Bits 1:0 – FIFOThreshold[1:0] FIFO level below which the module leaves the idle state to refill the FIFO, expressed in number of 128bit blocks.

43.8.31 FIFO depth register.

Name: RNG_Control_Registers_FIFODepth
Offset: 0x1100C
Reset: 0x00000008
Property: R

Table 43-32. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	FIFODepth[31:24]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	FIFODepth[23:16]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	FIFODepth[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FIFODepth[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	1	0	0	0

Bits 31:0 – FIFODepth[31:0] Maximum number of 32 bits words that can be stored in the FIFO:
 $2^{**g\_fifodepth}$.

43.8.32 Key register (MSB).

Name: RNG_Control_Registers_Key0
Offset: 0x11010
Reset: 0x00000000
Property: RW

Table 43-33. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	Key0[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	Key0[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	Key0[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	Key0[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Key0[31:0] Key register (MSB).

43.8.33 Key register.

Name: RNG_Control_Registers_Key1
Offset: 0x11014
Reset: 0x00000000
Property: RW

Table 43-34. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	Key1[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	Key1[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	Key1[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	Key1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Key1[31:0] Key register.

43.8.34 Key register.

Name: RNG_Control_Registers_Key2
Offset: 0x11018
Reset: 0x00000000
Property: RW

Table 43-35. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	Key2[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	Key2[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	Key2[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	Key2[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Key2[31:0] Key register.

43.8.35 Key register (LSB).

Name: RNG_Control_Registers_Key3
Offset: 0x1101C
Reset: 0x00000000
Property: RW

Table 43-36. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	Key3[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	Key3[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	Key3[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	Key3[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – Key3[31:0] Key register (LSB).

43.8.36 Test data register.

Name: RNG_Control_Registers_TestData
Offset: 0x11020
Reset: 0x00000000
Property: W

Table 43-37. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	TestData[31:24]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	TestData[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	TestData[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	TestData[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – TestData[31:0] Test data register.

43.8.37 Repetition Test Count Cut-Off value.

Name: RNG_Control_Registers_RepeatThreshold
Offset: 0x11024
Reset: 0x00000029
Property: RW

Table 43-38. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
			RepeatThreshold[5:0]					
Access			RW	RW	RW	RW	RW	RW
Reset			1	0	1	0	0	1

Bits 5:0 – RepeatThreshold[5:0] Repetition Test Count Cut-Off value.

43.8.38 Adaptive Proportion Test (1024-sample window) Cut-Off value.

Name: RNG_Control_Registers__PropThreshold
Offset: 0x11028
Reset: 0x00000319
Property: RW

Table 43-39. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							PropThreshold[9:8]	
Access							RW	RW
Reset							1	1
Bit	7	6	5	4	3	2	1	0
	PropThreshold[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	1	1	0	0	1

Bits 9:0 – PropThreshold[9:0] Adaptive Proportion Test (1024-sample window) Cut-Off value.

43.8.39 Status register.

Name: RNG_Control_Registers_Status
Offset: 0x11030
Reset: 0x00000000
Property: RW

Table 43-40. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
					FifoAccFail	StartUpFail	AlmInt	PreInt
Access					RW	R	RW	RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FullInt		PropFail	RepFail	State[2:0]		TestDataBusy	
Access	RW		RW	RW	R	R	R	R
Reset	0		0	0	0	0	0	0

Bit 11 – FifoAccFail Set when a FIFO data read is performed while the NDRNG is disabled AND has its FIFO empty (FIFOLevel = 0).

Bit 10 – StartUpFail Start-up test failure.

Bit 9 – AlmInt AIS31 noise alarm interrupt status.

Bit 8 – PreInt AIS31 preliminary noise alarm interrupt status.

Bit 7 – FullInt FIFO full status.

Bit 5 – PropFail NIST-800-90B adaptive Proportion Test (1024-sample window) interrupt status.

Bit 4 – RepFail NIST-800-90B repetition Count Test interrupt status.

Bits 3:1 – State[2:0] State of the control FSM: 000: Reset, 001: Startup, 010: Idle (Rings On), 011: Idle (Rings Off), 100: Fill FIFO, 101: Error, 110: Unused, 111: Unused.

Bit 0 – TestDataBusy High when data written to TestData register is being processed. (see section 4.7)

43.8.40 Initial wait counter value.

Name: RNG_Control_Registers_InitWaitVal
Offset: 0x11034
Reset: 0x0000FFFF
Property: RW

Table 43-41. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	InitWaitVal[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	InitWaitVal[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1

Bits 15:0 – InitWaitVal[15:0] Number of clock cycles to wait before sampling data from the noise source.

43.8.41 Disable oscillator rings #0 to #31.

Name: RNG_Control_Registers_DisableOsc0
Offset: 0x11038
Reset: 0x00000000
Property: RW

Table 43-42. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DisableOsc0[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DisableOsc0[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DisableOsc0[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DisableOsc0[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DisableOsc0[31:0] Disable oscillator rings #0 to #31.

43.8.42 Disable oscillator rings #32 to #63.

Name: RNG_Control_Registers_DisableOsc1
Offset: 0x1103C
Reset: 0x00000000
Property: RW

Table 43-43. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DisableOsc1[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	DisableOsc1[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	DisableOsc1[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	DisableOsc1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – DisableOsc1[31:0] Disable oscillator rings #32 to #63.

43.8.43 Switch off timer value.

Name: RNG_Control_Registers__SwOffTmrVal
Offset: 0x11040
Reset: 0x0000FFFF
Property: RW

Table 43-44. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	SwOffTmrVal[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1
Bit	7	6	5	4	3	2	1	0
	SwOffTmrVal[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	1	1	1	1	1	1	1	1

Bits 15:0 – SwOffTmrVal[15:0] Number of clk cycles to wait before stopping the rings after the FIFO is full.

43.8.44 Sample clock divider.

Name: RNG_Control_Registers_ClkDiv
Offset: 0x11044
Reset: 0x00000000
Property: RW

Table 43-45. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	ClkDiv[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – ClkDiv[7:0] Sample clock divider. The frequency at which the outputs of the rings are sampled is given by: $F_s = F_{clk} / (ClkDiv + 1)$.

43.8.45 AIS31 configuration register 0.

Name: RNG_Control_Registers__AIS31Conf0
Offset: 0x11048
Reset: 0x00000000
Property: RW

Table 43-46. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	OnlineThreshold[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	OnlineThreshold[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	StartUpThreshold[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	StartUpThreshold[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – OnlineThreshold[15:0] Online threshold.

Bits 15:0 – StartUpThreshold[15:0] Start-up test threshold.

43.8.46 AIS31 configuration register 1.

Name: RNG_Control_Registers__AIS31Conf1
Offset: 0x1104C
Reset: 0x00000000
Property: RW

Table 43-47. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	HEXpectedValue[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	HEXpectedValue[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	OnlineRepThreshold[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OnlineRepThreshold[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – HEXpectedValue[15:0] Expected history value.

Bits 15:0 – OnlineRepThreshold[15:0] Online repeat threshold.

43.8.47 AIS31 configuration register 2.

Name: RNG_Control_Registers__AIS31Conf2
Offset: 0x11050
Reset: 0x00000000
Property: RW

Table 43-48. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	HMax[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	HMax[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	HMin[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	HMin[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:16 – HMax[15:0] Maximum allowed history value.

Bits 15:0 – HMin[15:0] Minimum allowed history value.

43.8.48 AIS31 status register.

Name: RNG_Control_Registers_AIS31Status
Offset: 0x11054
Reset: 0x00000000
Property: RW

Table 43-49. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
							PrelimNoiseA larmRep	PrelimNoiseA larmRng
Access							RW	RW
Reset							0	0
Bit	15	14	13	12	11	10	9	8
	NumPrelimAlarms[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	NumPrelimAlarms[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bit 17 – PrelimNoiseAlarmRep Last preliminary noise alarm occurred due to consecutive high X**2.

Bit 16 – PrelimNoiseAlarmRng Last preliminary noise alarm occurred due to history value out of range.

Bits 15:0 – NumPrelimAlarms[15:0] Number of preliminary noise alarms since counter was last cleared.

43.8.49 Hardware configuration register.

Name: RNG_Control_Registers__HwConfig
Offset: 0x11058
Reset: 0x00000037
Property: R

Table 43-50. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
							AIS31Full	AIS31
Access							R	R
Reset							0	0
Bit	7	6	5	4	3	2	1	0
	NumOfRings[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	1	0	1	1	1

Bit 9 – AIS31Full Generic g_AIS31Full value.

Bit 8 – AIS31 Generic g_AIS31 value.

Bits 7:0 – NumOfRings[7:0] Generic g_NumRings value.

43.8.50 Pointers register.

Name: PK_Registers_Pointers
Offset: 0x12000
Reset: 0x00000000
Property: RW

Table 43-51. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
						OpPtrN[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	23	22	21	20	19	18	17	16
						OpPtrC[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	15	14	13	12	11	10	9	8
						OpPtrB[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0
Bit	7	6	5	4	3	2	1	0
						OpPtrA[3:0]		
Access					RW	RW	RW	RW
Reset					0	0	0	0

Bits 27:24 – OpPtrN[3:0] When executing primitive arithmetic operations, this pointer defines the location where the modulus is located in memory (location 0x0 to 0xF).

Bits 19:16 – OpPtrC[3:0] When executing primitive arithmetic operations, this pointer defines the location (0x0 to 0xF) where the result will be stored in memory.

Bits 11:8 – OpPtrB[3:0] When executing primitive arithmetic operations, this pointer defines where operand B is located in memory (location 0x0 to 0xF).

Bits 3:0 – OpPtrA[3:0] When executing primitive arithmetic operations, this pointer defines where operand A is located in memory (location 0x0 to 0xF).

43.8.51 Command register.

Name: PK_Registers_Command
Offset: 0x12004
Reset: 0x0000000F
Property: RW

Table 43-52. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	CalcR2	FlagB	FlagA	SwapBytes		Edwards	RandProj	RandKE
Access	RW	RW	RW	RW		RW	RW	RW
Reset	0	0	0	0		0	0	0
Bit	23	22	21	20	19	18	17	16
		SelCurve[2:0]			RandMod		OpBytesM1[9:8]	
Access		RW	RW	RW	RW		RW	RW
Reset		0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8
	OpBytesM1[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	FieldF	OpeAddr[6:0]						
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	1	1	1	1

Bit 31 – CalcR2 This bit indicates if the IP has to calculate $R^{*2} \bmod N$ for the next operation. This bit must be set to '1' when a new prime number has been programmed. This bit is used for primitive operations and ignored for the other operations. '0': don't recalculate $R^{*2} \bmod N$ '1': re-calculate $R^{*2} \bmod N$

Bit 30 – FlagB See operation description.

Bit 29 – FlagA See operation description.

Bit 28 – SwapBytes Swap the bytes on AHB interface: '0': Native format (little endian) '1': Byte swapped (big endian) See also section 4.2.1. This bit must be programmed before writing/reading any data in data memory.

Bit 26 – Edwards Enable Edwards curve (see operation description).

Bit 25 – RandProj Enable randomization of projective coordinates (counter-measure).

Bit 24 – RandKE Enable randomization of exponent/scalar (counter-measure).

Bits 22:20 – SelCurve[2:0] Enable accelerator for specific curve modulus: 0x0: No acceleration (default) 0x1: P256 0x2: P384 0x3: P521 0x4: P192 0x5: Curve25519 0x6: Ed25519 This field has no effect when the optional acceleration hardware is not included.

Bit 19 – RandMod Enable randomization of modulus (counter-measure).

Bits 17:8 – OpBytesM1[9:0] This field defines the size (= number of bytes minus one) of the operands for the current operation. Possible values are limited by the maximum supported operand size. Examples: - 0x014 -> ECC on curve K-163 - 0x01F -> ECC on curve P-256 - 0x02F -> ECC on curve P-384 - 0x033 -> ECC on curve K-409 - 0x041 -> ECC on curve P-521 - 0x07F -> 1024-bit RSA - 0x09F -> 1280-bit RSA - 0x1FF -> 4096-bit RSA - 0x3FF -> 8192-bit RSA

Bit 7 – FieldF '0': Field is GF(p) '1': Field is GF(2**m)

Bits 6:0 – OpeAddr[6:0] This field defines the operation to be performed. See documentation for more details.

43.8.52 Command register.

Name: PK_Registers_Control
Offset: 0x12008
Reset: 0x00000000
Property: W

Table 43-53. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
							ClearIRQ	Start
Access							W	W
Reset							0	0

Bit 1 – ClearIRQ Writing a '1' clears the IRQ output.

Bit 0 – Start Writing a '1' starts the processing.

43.8.53 STATUS Register.

Name: PK_Registers__Status
Offset: 0x1200C
Reset: 0x00000000
Property: R

Table 43-54. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
				FailPtr[4:0]				
Access				R	R	R	R	R
Reset				0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
							IntrptStatus	PK_Busy
Access							R	R
Reset							0	0
Bit	15	14	13	12	11	10	9	8
	ErrorFlags[11:4]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	ErrorFlags[3:0]							
Access	R	R	R	R				
Reset	0	0	0	0				

Bits 28:24 – FailPtr[4:0] These bits indicate which data location generated the error flag. They are not available for all error flags.

Bit 17 – IntrptStatus This bit reflects the IRQ output value. It is set when the operation is finished. It is cleared when the CPU writes the bit 1 of Control Register.

Bit 16 – PK_Busy This bit reflects the BUSY output value. It is set when the operation starts and it is cleared when the operation is finished.

Bits 15:4 – ErrorFlags[11:0] These bits indicate an error condition. They are updated at the end of the operation. They are cleared when starting a new operation.

43.8.54 Timer register.

Name: PK_Registers_Timer
Offset: 0x12014
Reset: 0x00000000
Property: RW

Table 43-55. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	Timer[30:23]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	Timer[22:15]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	Timer[14:7]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	Timer[6:0]							
Access	RW	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	

Bits 31:1 – Timer[30:0] Number of core clock cycles.

43.8.55 Hardware configuration register.

Name: PK_Registers__HwConfig
Offset: 0x12018
Reset: 0x43F30200
Property: R

Table 43-56. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	DisableCM	DisableClrMem	DisableSMx				AHBMaster	X25519
Access	R	R	R				R	R
Reset	0	1	0				1	1

Bit	23	22	21	20	19	18	17	16
	P192	P521	P384	P256		ECC	BinaryField	PrimeField
Access	R	R	R	R		R	R	R
Reset	1	1	1	1		0	1	1

Bit	15	14	13	12	11	10	9	8
	NbMult[3:0]				MaxOpSize[11:8]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	0

Bit	7	6	5	4	3	2	1	0
	MaxOpSize[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 31 – DisableCM State of DisableCM input (high when counter-measures are disabled).

Bit 30 – DisableClrMem State of DisableClrMem input (high when automatic clear of the RAM after reset is disabled).

Bit 29 – DisableSMx State of DisableSMx input (high when SM2/SM9 operations are disabled).

Bit 25 – AHBMaster '0': Memory access through AHB Slave and internally in the PKE. '1': Memory access through AHB Master, outside the PKE.

Bit 24 – X25519 Support Curve25519/Ed25519 acceleration.

Bit 23 – P192 Support ECC P192 acceleration.

Bit 22 – P521 Support ECC P521 acceleration.

Bit 21 – P384 Support ECC P384 acceleration.

Bit 20 – P256 Support ECC P256 acceleration.

Bit 18 – ECC Support RAM error correction.

Bit 17 – BinaryField Support binary field.

Bit 16 – PrimeField Support prime field.

Bits 15:12 – NbMult[3:0] Number of multipliers: 0x0: 1 multiplier 0x1: 4 multipliers 0x2: 16 multipliers 0x4: 64 multipliers 0x8: 256 multipliers

Bits 11:0 – MaxOpSize[11:0] Maximum operand size (number of bytes).

43.8.56 Operand size register.

Name: PK_Registers__OpSize
Offset: 0x1201C
Reset: 0x00001000
Property: RW

Table 43-57. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
				OpSize[12:8]				
Access				RW	RW	RW	RW	RW
Reset				1	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	OpSize[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 12:0 – OpSize[12:0] Operand size (number of bytes): 0x0100: 256 0x0209: 521 0x0800: 2048 0x0C00: 3072 0x1000: 4096 This register is used when the memory is accessed via AHB Master

43.8.57 Memory offset register.

Name: PK_Registers_MemOffset
Offset: 0x12020
Reset: 0x00000000
Property: RW

Table 43-58. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MemOffset[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MemOffset[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MemOffset[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MemOffset[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MemOffset[31:0] Memory offset for AHB Master

43.8.58 MicroCode offset register.

Name: PK_Registers_MicroCodeOffset
Offset: 0x12024
Reset: 0x00000000
Property: RW

Table 43-59. Register Bit Attribute Legend

Symbol	Description	Symbol	Description	Symbol	Description
R	Readable bit	HC	Cleared by Hardware	(Grey cell)	Unimplemented
W	Writable bit	HS	Set by Hardware	X	Bit is unknown at Reset
K	Write to clear	S	Software settable bit	—	—

Bit	31	30	29	28	27	26	25	24
	MicroCodeOffset[31:24]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	MicroCodeOffset[23:16]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	MicroCodeOffset[15:8]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
	MicroCodeOffset[7:0]							
Access	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – MicroCodeOffset[31:0] MicroCode offset for AHB Master

44. Anti-Tamper Module (AT)

44.1 Overview

The Anti-Tamper (AT) module manages tamper events by generating responses (interrupt, erase tamper key, erase Flash key) based on the tamper level configuration for the tamper source. It supports 30 tamper source inputs. The AT module is integrated into the backup domain to support operation in Backup and Hibernate Low-Power modes.

45. Physically Uncloneable Function (PUF)

45.1 Overview

The Physically Unclonable Function (PUF) generates device unique keys that can be used for local encryption and attestation functions.

46. Electrical Characteristics

46.1 Absolute Maximum Electrical Characteristics

Absolute maximum ratings for are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these or any other conditions, above the parameters indicated in the operation listings of this specification, is not implied.

Table 46-1. Absolute Maximum Electrical Characteristics

Ambient temperature under bias	-40°C to +85°C
Storage temperature	-65°C to 150°C
Voltage on VDD with respect to VSS	-0.3V to 4.03V
Voltage on any pin that is not 5V tolerant, with respect to VSS ⁽³⁾	-0.3V to (VDDIO+0.3V)
Voltage on any 5V tolerant pin with respect to VSS when VDDIO ≥ 1.9V ⁽³⁾	-0.3V to 5.5V
Voltage on any 5V tolerant pin with respect to VSS when VDDIO < 1.9V ⁽³⁾	-0.3V to (VDDIO+0.3V)
Voltage on USBFS D+ or D- pin with respect to VSS	-0.3V to (VDDIO+0.3V)
Voltage on VREFH with respect to VSS	-0.3V to +(AVDD+0.3V)
Maximum current out of a VSS pins ⁽²⁾	200 mA
Maximum current into a VDDIO pins ⁽²⁾	200 mA
Maximum output current sourced/sunk by any 4x I/O pin	15 mA
Maximum output current sourced/sunk by any 8x I/O pin	25 mA
Maximum output current sunk/sourced by any port cluster ⁽⁴⁾	200 mA
Maximum Junction Temperature	+105°C
ESD qualification:	
Human Body Model (HBM) per JESD22-A114	(*)
Charged Device Model (CDM) (ANSI/ESD STM 5.3.1)	(*)
Notes:	
1. Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions, above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.	
2. Maximum allowable current is a function of device maximum power dissipation .	
3. When applying higher or lower voltage than those specified on IO pins, please refer to DI_19 / DI_21 to respect the maximum injection current specification.	
4. Refer to table 46-17 for the I/O pin Drive Strength.	
5. This maximum value is specific to continuous DC current. For IO pin AC current characteristics, refer to the 'I/O Pin Electrical Specifications'.	
6. A cluster is a group of IO pins bounded by power/ground pins. Refer to the following table 'I/O Clusters'.	

Table 46-2. I/O Clusters

IO Cluster	Pins
1	PD06,PD07,PD19,PD04,PD05,PD08,PD20,PD09,PA00,PA14,PA01,PA15,PA02
2	PA10,PA11,PA12,PA16,PA17,PA18,PA03,PA04,PA05,PA06,PA13
3	PA07,PA08,PA09,PA20,PA19,PB00,PB07,PB01,PB02
4	PB11,PB12,PB13,PB14,PB03,PB04,PB05,PB08,PB09,PB06
5	PB15,PB16,PB17,PB10, PC00,PC11,PC12,PC01,PC13
6	PC02,PC14,PC15,PC03,PC16,PC09,PC04,PC17,PC10
7	PC18,PC19,PC05,PC06,PC07,PC08,PD00,PC10,PC14, PD11,PD01
8	PD02,PD03,PD15,PD16,PD17,PD12,PD18,PD13

46.2 CPU Electrical Characteristics

Table 46-3. Operating Frequency vs Voltage

Param. No.	VDDIOx, AVDD Range	Temp. Range (in °C)	Max MCU Frequency	Comments
DC_5	1.71 to 3.63V	-40°C to +85°C	FCLK_1 MHz	Industrial

Table 46-4. MCU Thermal Operating Conditions

Rating	Symbol	Min.	Typ.	Max.	Unit
Industrial Temperature Devices Operating Ambient Temperature Range Operating Junction Temperature Range	TA TJ	-40 —	—	85 105	°C °C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = (VDDIOx \times (IDD - \sum IOHVDDIOx)) + (AVDD \times (IDDANA - \sum IOHAVDD)) + (VDDREG \times IDDREG)$ I/O Pin Power Dissipation: $PI/O = \sum ((VDDIOx - VOHVDDIOx) \times IOHVDDIOx) + \sum (VOL \times IOLVDDIOx) + \sum ((AVDD - VOHAVDD) \times IOHAVDD) + \sum (VOL \times IOLAVDD)$	PD	PINT + PI/O			W
Maximum Allowed Power Dissipation	PDMAX	(TJ - TA)/θJA			W

Table 46-5. Thermal Packaging Characteristics ⁽¹⁾

Characteristics	Symbol	Typ.	Max.	Unit
Thermal Resistance, 48-pin TQFP (7x7x1 mm) Package	θJA	47.0	—	°C/W
Thermal Resistance, 64-pin TQFP (10x10x1 mm) Package	θJA	44.8	—	°C/W
Thermal Resistance, 100-pin TQFP (14x14x1 mm) Package	θJA	41.1	—	°C/W
Thermal Resistance, 48-pin VQFN (7x7x0.9 mm) Package	θJA	22.3	—	°C/W
Thermal Resistance, 64-pin VQFN (9x9x1 mm) Package	θJA	20.5	—	°C/W

Note:

1. Junction to ambient thermal resistance, Theta-JA (θJA) numbers are achieved by package simulations.

46.3 Power Supply

Table 46-6. Power Supply DC Electrical Specifications

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
REG_5	VDDIO_CIN ⁽⁴⁾	VDDIO Input Bypass parallel Capacitor pair	33	—	—	μF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω ⁽⁵⁾
			100	—	—	nF	Ceramic X7R with ESR <0.5Ω on all VDDIOx pins ⁽⁵⁾
REG_6	VDDREG_CIN ⁽⁴⁾	VDDREG Input Bypass parallel Capacitor pair	33	—	—	μF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω ⁽⁵⁾
			100	—	—	nF	Ceramic X7R with ESR <0.5Ω on all VDDIOx pins ⁽⁵⁾
REG_15	USERLDO_COUT	User LDO output capacitor	10	—	—	μF	Ceramic X7R with ESR <0.5Ω
REG_17	AVDD_CIN ⁽⁴⁾	AVDD Input Bypass parallel Capacitor pair	10	—	—	μF	Bulk Ceramic or solid Tantalum with ESR <0.5Ω
			100	—	—	nF	Ceramic X7R with ESR <0.5Ω

Table 46-6. Power Supply DC Electrical Specifications (continued)

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
REG_23	AVDD_LEXT ⁽¹⁾	AVDD series Ferrite Bead DCR (DC Resistance)	—	—	0.1	Ω	≥600 Ohms @ 100 MHz
REG_25		Ferrite Bead current Rating	500	—	—	mA	—
REG_37	VDDIO ⁽²⁾	VDDIO Input Voltage Range	1.71	3.3	3.63	V	—
REG_39	AVDD ⁽²⁾	AVDD Input Voltage Range	1.71	3.3	3.63	V	—
REG_40	VDDREG ⁽³⁾	VDDREG Input Voltage Range	1.71	3.3	3.63	V	—
REG_43	SVDD_R	VDDIO/AVDD/VDDREG Rise Ramp Rate to Ensure Internal Power-on Reset Signal	0.00000033	—	0.33	V/μs	Failure to meet this specification may lead to start-up or unexpected behaviors
REG_45	VPOR	Power-on Reset	1.44	1.5	1.59	V	VDD Power-down
REG_47	VDDIO / AVDD BOR ⁽⁵⁾	VDDIO / AVDD Brown-out Reset Thresholds	(*)	—	(*)	V	BOR_TRIP_VDDx = 0x0 HYST_BOR_VDDx = 0x0
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x1 HYST_BOR_VDDx = 0x0
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x2 HYST_BOR_VDDx = 0x0
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x3 HYST_BOR_VDDx = 0x0
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x0 HYST_BOR_VDDx = 0x1
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x1 HYST_BOR_VDDx = 0x1
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x2 HYST_BOR_VDDx = 0x1
			(*)	—	(*)	V	BOR_TRIP_VDDx = 0x3 HYST_BOR_VDDx = 0x1
REG_49	VDDREG BOR ⁽⁵⁾	VDDREG Brown-out Reset Thresholds	(*)	—	(*)	V	HYST_BOR_VDDREG = 0x0 ⁽⁶⁾
			(*)	—	(*)	V	HYST_BOR_VDDREG = 0x1 ⁽⁶⁾

Table 46-6. Power Supply DC Electrical Specifications (continued)

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
REG_51	VDDIO / AVDD / VDDREG DCBOR ⁽⁷⁾	VDDIO / AVDD / VDDREG Duty Cycled BOR Thresholds	1.43	—	1.75	V	BOR_TRIP = 0x0 BOR_HYS = 0x0
			1.87	—	2.28	V	BOR_TRIP = 0x1 BOR_HYS = 0x0
			2.22	—	2.75	V	BOR_TRIP = 0x2 BOR_HYS = 0x0
			2.43	—	3.01	V	BOR_TRIP = 0x3 BOR_HYS = 0x0
			1.4	—	1.75	V	BOR_TRIP = 0x0 BOR_HYS = 0x1
			1.81	—	2.28	V	BOR_TRIP = 0x1 BOR_HYS = 0x1
			2.13	—	2.75	V	BOR_TRIP = 0x2 BOR_HYS = 0x1
			2.27	—	3.01	V	BOR_TRIP = 0x3 BOR_HYS = 0x1
REG_52	LVD ⁽⁸⁾	VDDIO Low Voltage Detector Thresholds	(*)	—	(*)	V	LVD.LEVEL = 0x0
			(*)	—	(*)	V	LVD.LEVEL = 0x1
			(*)	—	(*)	V	LVD.LEVEL = 0x2
			(*)	—	(*)	V	LVD.LEVEL = 0x3
			(*)	—	(*)	V	LVD.LEVEL = 0x4
			(*)	—	(*)	V	LVD.LEVEL = 0x5
			(*)	—	(*)	V	LVD.LEVEL = 0x6
			(*)	—	(*)	V	LVD.LEVEL = 0x7
			(*)	—	(*)	V	LVD.LEVEL = 0x8
			(*)	—	(*)	V	LVD.LEVEL = 0x9
			(*)	—	(*)	V	LVD.LEVEL = 0xA
			(*)	—	(*)	V	LVD.LEVEL = 0xB
			(*)	—	(*)	V	LVD.LEVEL = 0xC
			(*)	—	(*)	V	LVD.LEVEL = 0xD
			(*)	—	(*)	V	LVD.LEVEL = 0xE
			(*)	—	(*)	V	LVD.LEVEL = 0xF
REG_53	TRST	External RESET valid active pulse width	2	—	—	μs	Minimum reset active time to guarantee MCU reset

Table 46-6. Power Supply DC Electrical Specifications (continued)

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. Ferrite Bead ISAT(min) ≥ (IDDANA(max) * 1.15).
2. VDDREG voltage must be equal or lower than VDDIO.
3. VDDIO and AVDD must be at the same voltage level.
4. All bypass caps should be located immediately adjacent to pins and on the same side of the PCB as the MCU. Each primary power supply group VDDIO, AVDD, VDDREG should have one bulk capacitor and all power pins everywhere a 100 nF bypass cap.
5. Voltages below the minimum BOR threshold will result in a device reset, except for the BOR_VDDUSB that can cause an interrupt. Voltages above the maximum BOR threshold will allow the device starting-up.
6. Overall functional device operation at VBORMIN < VDD < VDDMIN is guaranteed, but not characterized. Device will function with degraded performances below VDDMIN.
7. Voltages below the minimum DCBOR threshold will result in a device reset.
8. Voltages below the minimum LVD threshold will result in a falling detection. Voltages above the maximum LVD threshold will result in a rising detection. Enabling the LVD when VDDIO is between the minimum or maximum threshold may result in unexpected behavior.

46.4 Active Power

Table 46-7. MCU Active Current Consumption DC Electrical Specifications

DC CHARACTERISTICS					Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Clock/Freq	Power Pin	Typ. ⁽¹⁾	Max.	Units	Conditions
APWR_1	IDD_ACTIVE ^(2,3)	MCU IDD in Active mode	PLL_72_MHz	VDDIO = AVDD = 3.3V	2.6	(*)	mA	PLL runs at 72 MHz, uses XOSC with 48 MHz crystal as reference clock
APWR_3				VDDREG = 3.3V	11.1	(*)	mA	
APWR_5			DFLL_48_MHz	VDDIO = AVDD = 3.3V	0.8	(*)	mA	DFLL in open loop
APWR_7				VDDREG = 3.3V	4.9	(*)	mA	
APWR_9			XOSC_48_MHz	VDDIO = AVDD = 3.3V	1.7	(*)	mA	48 MHz crystal
APWR_11				VDDREG = 3.3V	5.4	(*)	mA	

Table 46-7. MCU Active Current Consumption DC Electrical Specifications (continued)

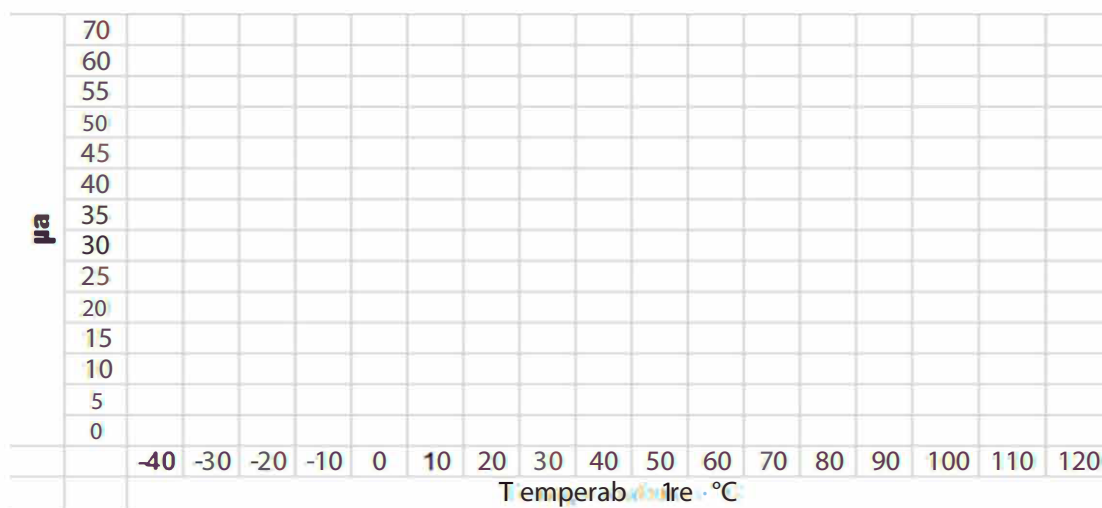
DC CHARACTERISTICS					Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Clock/Freq	Power Pin	Typ. ⁽¹⁾	Max.	Units	Conditions
Notes: 1. Typical values at 25°C only. 2. Conditions: – RTC running on ULP32K for CoreMark® ticks count – No other peripheral modules are operating (i.e., all peripherals except RTC inactive) – APB Peripheral bus clocks: default settings after reset – MCU is running on Flash with automatic wait state – I/Os are inactive input mode with input trigger disabled – All clock generation sources disabled unless otherwise specified – WDT, CFD Clock Fail Detect disabled 3. MCU Running CoreMark Test Suite.								

46.5 Idle Power

Table 46-8. MCU Idle Current Consumption DC Electrical Specifications

DC CHARACTERISTICS					Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Clock/Freq	Power Pin	Typ. ⁽¹⁾	Max.	Units	Conditions
IPWR_1	IDD_IDLE ⁽²⁾	MCU IDD in Idle mode	PLL_72_MHz	VDDIO = AVDD = 3.3V	1.4	(*)	mA	PLL runs at 72 MHz, uses XOSC with 48 MHz crystal as reference clock
IPWR_3				VDDREG = 3.3V	5.6	(*)	mA	
IPWR_5			DFLL_48_MHz	VDDIO = AVDD = 3.3V	0.8	(*)	mA	DFLL in open loop
IPWR_7				VDDREG = 3.3V	1.8	(*)	mA	
IPWR_9			XOSC_48_MHz	VDDIO = AVDD = 3.3V	1.4	(*)	mA	48 MHz crystal
IPWR_11				VDDREG = 3.3V	2.1	(*)	mA	
Notes: 1. Typical values at 25°C only 2. Conditions: - No peripheral modules are operating (i.e., all peripherals inactive) - APB Peripheral bus clocks: default settings after reset - MCU is running on Flash with automatic wait state - I/Os are inactive input mode with input trigger disabled - All clock generation sources disabled unless otherwise specified - WDT, CFD Clock Fail Detect disabled								

Figure 46-1. Power Consumption over Temperature in Idle mode (Typical values for guidance only, not characterized over process or voltage).



Operating Conditions:

- VDDIO = 3.3V
- PLL_72_MHz

46.6 Standby Power

Table 46-9. MCU Standby Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Power Pin	Typ. (1)	Max.	Units	Conditions
SPWR_1	IDD_STANDBY (2)	MCU IDD in Standby mode w/VDDCORE_x = 0.8V	VDDIO = AVDD = 3.3V	0.22	(*)	mA	RAM Low Power mode disabled (PM.STDBYCFG.LPRAM = 0x0), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1) Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0)
SPWR_3			VDDREG = 3.3V	0.47	(*)	mA	
SPWR_5			VDDIO = AVDD = 3.3V	0.22	(*)	mA	RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1) Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0)
SPWR_7			VDDREG = 3.3V	0.47	(*)	mA	
SPWR_9			VDDIO = VDDANA = 3.3V	0.13	(*)	mA	RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1) Voltage Reference Low Voltage enabled (SUPC.VREF.LPSTDBY = 0x1)
SPWR_11			VDDREG = 3.3V	0.16	(*)	mA	
SPWR_13		MCU IDD in Standby mode w/VDDCORE_x = 1.2V	VDDIO = AVDD = 3.3V	0.25	(*)	mA	RAM Low-Power mode disabled (PM.STDBYCFG.LPRAM = 0x0), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0) Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0)
SPWR_15			VDDREG = 3.3V	0.7	(*)	mA	
SPWR_17			VDDIO = AVDD = 3.3V	0.25	(*)	mA	RAM Low-Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0) Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0)
SPWR_19			VDDREG = 3.3V	0.7	(*)	mA	
SPWR_21			VDDIO = AVDD = 3.3V	0.16	(*)	mA	RAM Low- Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0) Voltage Reference Low Voltage enabled (SUPC.VREF.LPSTDBY = 0x1)
SPWR_23			VDDREG = 3.3V	0.33	(*)	mA	

Table 46-9. MCU Standby Current Consumption DC Electrical Specifications (continued)

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Power Pin	Typ. (1)	Max.	Units	Conditions
Notes: 1. Typical values at 25°C only. 2. Conditions: - No peripheral modules are operating, unless specified (i.e., all peripherals inactive) - APB Peripheral bus clocks: default settings after reset - MCU is running on Flash with automatic wait state - I/Os are inactive input mode with input trigger disabled - All clock generation sources disabled unless otherwise specified - WDT, CFD Clock Fail Detect disabled							

46.7 Hibernate Power

Table 46-10. MCU Hibernate Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Power Pin	Typ. ⁽¹⁾	Max.	Units	Conditions
HPWR_1	IDD_HIBERNATE ⁽²⁾	MCU IDD in Hibernate mode w/VDDCORE_x = 0.8V	VDDIO = AVDD = 3.3V	100	(*)	μA	RAM Low- Power mode disabled (PM.HIBCFG.LPRAM = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_3			VDDREG = 3.3V	200	(*)	μA	Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)
HPWR_5			VDDIO = AVDD = 3.3V	100	(*)	μA	RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_7			VDDREG = 3.3V	200	(*)	μA	Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)
HPWR_8			VDDIO = VDDANA = 3.3V	101	(*)	μA	RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_9			VDDREG = 3.3V	200	(*)	μA	Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0) RTC running on XOSC32K
HPWR_10			VDDIO = AVDD = 3.3V	18	(*)	μA	RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_11			VDDREG = 3.3V	47	(*)	μA	Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1)
HPWR_12			VDDIO = AVDD = 3.3V	19	(*)	μA	RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)
HPWR_13			VDDREG = 3.3V	48	(*)	μA	Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1) RTC running on XOSC32K

Table 46-10. MCU Hibernate Current Consumption DC Electrical Specifications (continued)

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Power Pin	Typ. ⁽¹⁾	Max.	Units	Conditions
HPWR_14	IDD_HIBERNATE ⁽²⁾	MCU IDD in Hibernate mode w/VDDCORE_x = 1.2V	VDDIO = AVDD = 3.3V	100	(*)	μA	RAM Low- Power mode disabled (PM.HIBCFG.LPRAM = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)
HPWR_15			VDDREG = 3.3V	294	(*)	μA	Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)
HPWR_16			VDDIO = AVDD = 3.3V	100	(*)	μA	RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)
HPWR_17			VDDREG = 3.3V	292	(*)	μA	Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)
HPWR_18			VDDIO = AVDD = 3.3V	101	(*)	μA	RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)
HPWR_19			VDDREG = 3.3V	292	(*)	μA	Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0) RTC running on XOSC32K
HPWR_20			VDDIO = AVDD = 3.3V	18	(*)	μA	RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)
HPWR_21			VDDREG = 3.3V	101	(*)	μA	Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1)
HPWR_22			VDDIO = AVDD = 3.3V	19	(*)	μA	RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)
HPWR_23			VDDREG = 3.3V	101	(*)	μA	Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1) RTC running on XOSC32K

Notes:

- Typical values at 25°C only.
- Conditions:
 - No peripheral modules are operating, unless specified (i.e., all peripherals inactive)
 - APB Peripheral bus clocks: default settings after reset
 - MCU is running on Flash with automatic wait state
 - I/Os are inactive input mode with input trigger disabled
 - All clock generation sources disabled unless otherwise specified
 - WDT, CFD Clock Fail Detect disabled

46.8 Backup Power

Table 46-11. MCU Backup Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Power Pin	Typ ⁽¹⁾	Max.	Units	Conditions
BPWR_1	IDD_BACKUP ⁽²⁾	MCU IDD in Backup mode powered from VDDIOx	VDDIO = AVDD = 3.3V	4.8	(*)	μA	RTC running on XOSC32K (32.768 KHz), TRAM enabled
BPWR_3			VDDREG = 3.3V	0.6	(*)	μA	

Notes:

- Typical values at 25°C only.
- Conditions:
 - No peripheral modules are operating (i.e., all peripherals inactive)
 - APB Peripheral bus clocks: default settings after reset
 - CMCC Cache disabled
 - MCU is running on Flash with automatic wait state
 - I/Os are inactive input mode with input trigger disabled
 - All clock generation sources disabled unless otherwise specified
 - WDT, CFD Clock Fail Detect disabled

46.9 OFF Power

Table 46-12. MCU Off Current Consumption DC Electrical Specifications

DC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Power Pin	Typ. ⁽¹⁾	Max.	Units	Conditions
OPWR_1	IDD_OFF ⁽²⁾	MCU IDD in OFF	VDDIO = AVDD = 3.3V	1	(*)	μA	In Off mode, the device is entirely powered-off. (SLEEP_CFG.SLEEP_MODE = OFF), and subsequent execution of the WFI instruction. Note: This mode is left by pulling the RESET pin low, or when a power Reset is done.
OPWR_3			VDDREG = 3.3V	0.03	(*)	μA	

Notes:

- Typical values at 25°C only.
- Conditions:
 - All peripherals inactive unless otherwise specified
 - All clock generation sources disabled unless otherwise specified
 - All I/O pins configured as input pins pulled down or tied to GND

46.10 Wake-Up Timing

Table 46-13. Wake-Up Timings from Low-Power Modes AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
WKUP_1	WKUP_IDLE	Wake from Idle mode	—	0.311	—	μs	
WKUP_7	WKUP_STDBY	Wake from Stndby mode	—	12.5	—	μs	
WKUP_11	WKUP_HIBER	Wake from Hibernate mode	—	1.66	—	ms	
WKUP_13	WKUP_BCKUP	Wake from Backup mode	—	1.63	—	ms	
WKUP_15	WKUP_OFF	Wake from Off mode	—	1.40	—	ms	
Note:							
1. VDD = 3.3V, CPU clock = 48 MHz, automatic wait states, Temperature = 25°C.							

46.11 Peripheral Active Power

Table 46-14. Peripheral Active Current DC Electrical Specifications

DC CHARACTERISTICS ⁽¹⁾			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial
Param. No.	Symbol	Characteristics	Conditions
MODULES/PERIPHERALS ACTIVE CURRENTS ≤1mA			
PAI_51	AC	Analog Comparator Current	2 Comparators Enabled
PAI_52	BOR	BOR Current	—
PAI_53	CCL	CCL Current	—
PAI_54	EIC	EIC Current	—
PAI_55	EVSYS	EVSYS Current	—
PAI_56	FREQM	FREQM Current	—
PAI_57	SERCOM	SERCOM Current	SPI mode, internal loopback
PAI_58	WDT	WDT Current	—
MODULES/PERIPHERALS ACTIVE CURRENTS ≤3mA			
PAI_100	ADC	Analog-to- Digital Converter Current	12-bit, 4.5 Msps
PAI_101	CAN-FD	CAN-FD Current	8 Mbps 10 Mbps
PAI_102	PTC	PTC Current	—
PAI_103	TCC	TCC Current	48 MHz
PAI_104	USB FS	Full-Speed USB Current	—
PAI_106	XOSC	XOSC Current	48 MHz crystal, AGC Enabled
MODULES/PERIPHERALS ACTIVE CURRENTS ≤10mA			
PAI_400	DMA	DMA Current	—
PAI_401	Flash	Flash Erase/Write	—
PAI_402	PLL	PLL Active Current	PLL output = 72 MHz

Table 46-14. Peripheral Active Current DC Electrical Specifications (continued)

DC CHARACTERISTICS ⁽¹⁾			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Conditions				

Note:

- These values are for worst case guidance only unless otherwise specified, the actual values depend on user peripheral configuration and operating frequency.

46.12 I/O Pin Electrical Characteristics

Table 46-15. I/O Pin AC/DC Electrical Specifications

AC - DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
DI_1	VIL	Input Low Voltage I/O Pins	GND	—	0.2*VDDIO	V	VDDIO(min) -to- VDDIO(max)
DI_3	VIH	Input High Voltage Non-5v tolerant I/O Pins	0.7*VDDIO	—	VDDIO	V	VDDIO(min) -to- VDDIO(max)
		Input High Voltage I/O Pins 5V-tolerant	0.7*VDDIO	—	5.5V	V	VDDIO(min) -to- 5.5V
DI_5	VOL	4x Drive Strength I/O pins (Output Low)	—	—	(*)	V	VDDIO < 3.0V @ IOL = _____ ⁽⁶⁾
							VDDIO ≥ 3.0V @ IOL = _____ ⁽⁶⁾
		8x Drive Strength I/O pins (Output Low)					VDDIO < 3.0V @ IOL = _____ ⁽⁶⁾
							VDDIO ≥ 3.0V @ IOL = _____ ⁽⁶⁾
DI_9	VOH	4x Drive Strength I/O pins (Output High)	(*)	—	—	V	VDDIO < 3.0V @ IOL = _____ ⁽⁶⁾
							VDDIO ≥ 3.0V @ IOL = _____ ⁽⁶⁾
		8x Drive Strength I/O pins (Output High)					VDDIO < 3.0V @ IOL = _____ ⁽⁶⁾
							VDDIO ≥ 3.0V @ IOL = _____ ⁽⁶⁾
DI_13	IIL	Input pin leakage current	(*)	—	(*)	μA	GND ≤ VPIN ≤ VDDIO(max) (VPIN = Voltage present on Pin)
DI_14	IIL_5V	Input 5.5V tolerant pin leakage current	(*)	—	(*)	μA	For VDDIO < 1.9V: GND ≤ VPIN ≤ VDDIO+3.6 For VDDIO ≥ 1.9V: 1.9V ≤ VPIN ≤ 5.5V (VPIN = Voltage present on Pin)
DI_15	RPDWN	Internal Pull-Down (DIR = OUT = 0, PULLEN = 1)	(*)	—	(*)	kΩ	VDDIOx(min) - VDDIO(max)
DI_17	RPUP	Internal Pull-Up (DIR = 0, OUT = PULLEN = 1)	(*)	—	(*)	kΩ	

Table 46-15. I/O Pin AC/DC Electrical Specifications (continued)

AC - DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
DI_19	IICL	Input Low Injection Current	-5	—	—	mA	This parameter applies to all I/O pins (Notes 1, 4, 5).
DI_21	IICH	Input High Injection Current	—	—	8	mA	This parameter applies to all pins, with the exception of 5V tolerant I/O pins. (Notes 2, 3, 4, 5).
DI_23	ΣIICT	Total Input Injection Current (sum of all I/O and control pins) Absolute value of ΣIICT	-69	—	69	mA	Absolute instantaneous sum of all ± input injection currents from all I/O pins. (IICL + IICH) ≤ ΣIICT
DI_25	TRISE	I/O pin Rise Time (DRVSTR = 4x)	—	—	10	ns	VDDIOx(min), CLOAD = 30 pF(max)
DI_27	TFALL	I/O pin Fall Time (DRVSTR = 4x)	—	—	9	ns	
DI_29	TRISE	I/O pin Rise Time (DRVSTR = 8x, SLEW = FAST)	—	—	5.5	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW4)	—	—	21	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW8)	—	—	38	ns	
		I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW12)	—	—	54	ns	
DI_31	TFALL	I/O pin Fall Time (DRVSTR = 8x, SLEW = FAST)	—	—	5.5	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW4)	—	—	24	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW8)	—	—	43	ns	
		I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW12)	—	—	61	ns	

Table 46-15. I/O Pin AC/DC Electrical Specifications (continued)

AC - DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions

Notes:

- VIL source < (GND - 0.3). Characterized but not tested.
- VIH source > (VDDIO + 0.3) for non-5V tolerant pins only.
- Digital 5V tolerant pins do not have an internal high side Diode to VDDIO, therefore, cannot tolerate any 'positive' input injection current.
- If the sum of all injection currents are > | $\sum I_{ICT}$ | it can affect the ADC results by approximately 4 to 6 counts (i.e., VIH Source > (VDDIO + 0.3) or VIL source < (GND - 0.3)).
- Any number and the combination of I/O pins not excluded under IICL or IICH conditions are permitted provided the 'absolute instantaneous' sum of the input injection currents from all pins do not exceed the specified $\sum I_{ICT}$ limit. To limit the injection current the user must insert a resistor in series R_{SERIES}, (i.e., RS), between input source voltage and device pin. The resistor value is calculated according to:
 - For negative Input voltages less than (GND-0.3): $RS \geq \text{absolute value of } | (VIL \text{ source} - (GND - 0.3)) / I_{ICL} |$
 - For positive input voltages greater than (VDDIO +0.3): $RS \geq ((VIH \text{ source} - (VDDIO +0.3)) / I_{ICH})$
 - For Vpin voltages >VDDIO +0.3 and <GND-0.3 then RS = the larger of the values calculated above
- IOL/IOH is specific to AC operation. For continuous DC operation characteristics, see the [Absolute Maximum Ratings](#) section.
- Not all I²C operating modes are supported on all pin types. Refer to the following table, I²C Support.
- See the I/O Drive Strength table for the I/O pin Drive Strength.

Table 46-16. I²C Support

I ² C Mode	I/O Pin Drive Strength	
	4x	8x
I ² C Standard Mode (SM), 100 kHz w/400 pF load	No	Yes
I ² C Fast Mode (FM), 400 kHz w/400 pF load	No	Yes
I ² C Fast Mode Plus (FM+), 1 MHz w/550 pF load	No	Yes
I ² C High Speed (HS), 3.4 MHz w/100 pF load	No	Yes

Table 46-17. I/O Drive Strength

Drive Strength	IO Port/Pad
4x	PA10, PA11, PA12, PA13, PA07, PA08, PA09, PA20, PA19, PB00, PB07, PB01, PB02, PB11, PB12, PB13, PB14, PB08, PB09, PB10, PC09, PC10, PC07, PC08, PD02, PD03, PD12, PD13, PD04, PD05
8x	PD09, PA00, PA14, PA01, PA15, PA02, PA16, PA17, PA18, PA03, PA04, PA05, PA06, PB03, PB04, PB05, PB06, PB15, PB16, PB17, PC00, PC11, PC12, PC01, PC13, PC02, PC14, PC15, PC03, PC16, PC04, PC17, PC18, PC19, PC05, PC06, PD00, PD10, PD14, PD11, PD01, PD15, PD16, PD17, PD18, PD06, PD07, PD19, PD08, PD20

Figure 46-2. VOL vs IOL, I/O Low Drive Strength I/O Pin Plot

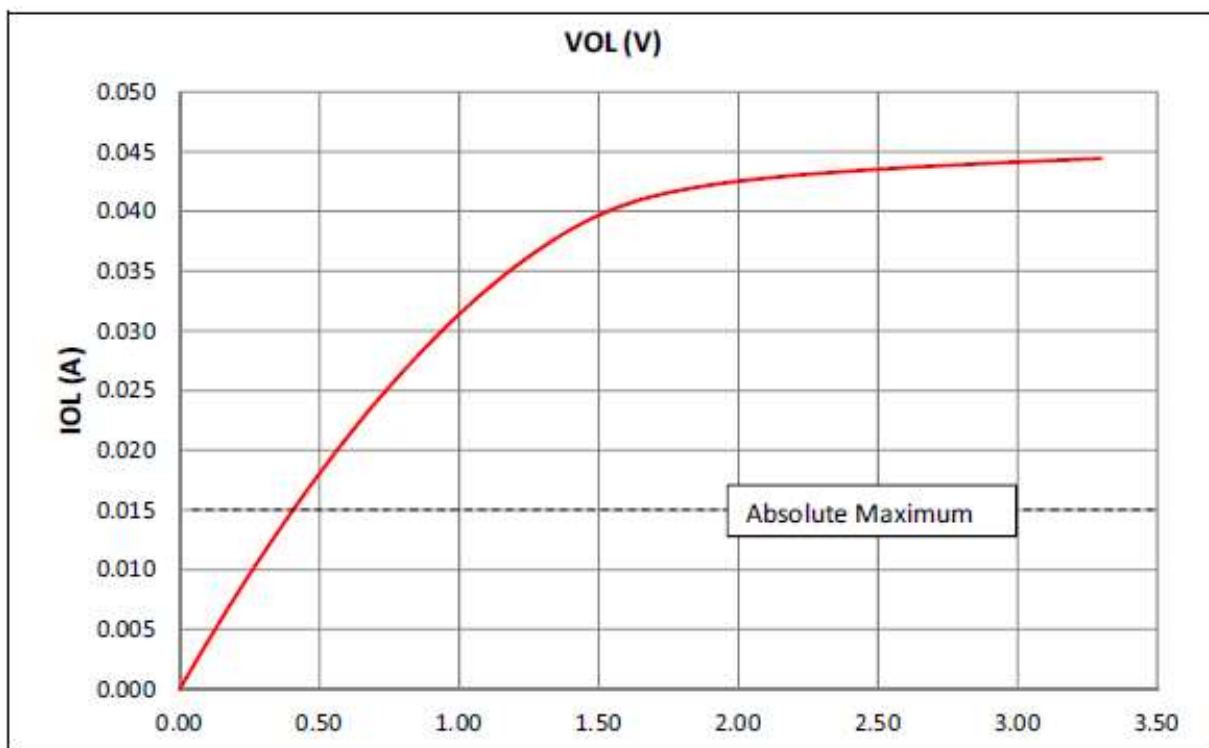


Figure 46-3. VOH vs IOH, I/O Low Drive Strength I/O Pin Plot

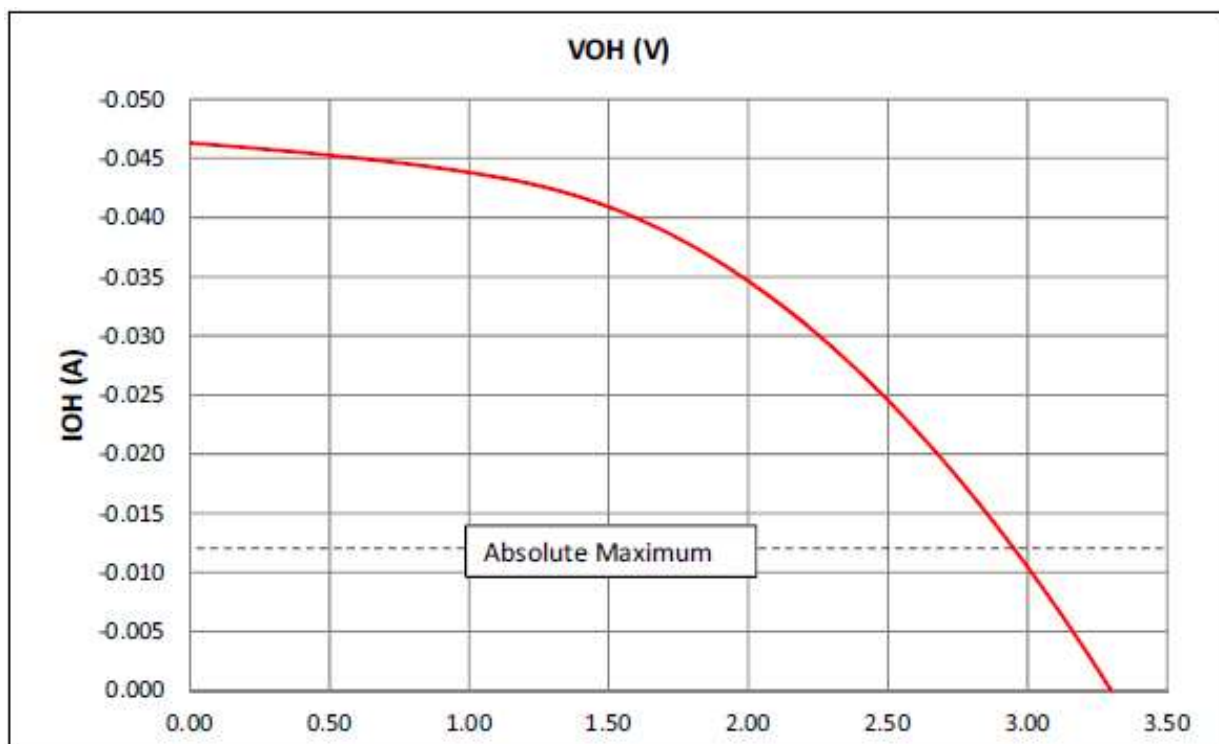


Figure 46-4. VOL vs IOL, I/O High Drive Strength I/O Pin Plot

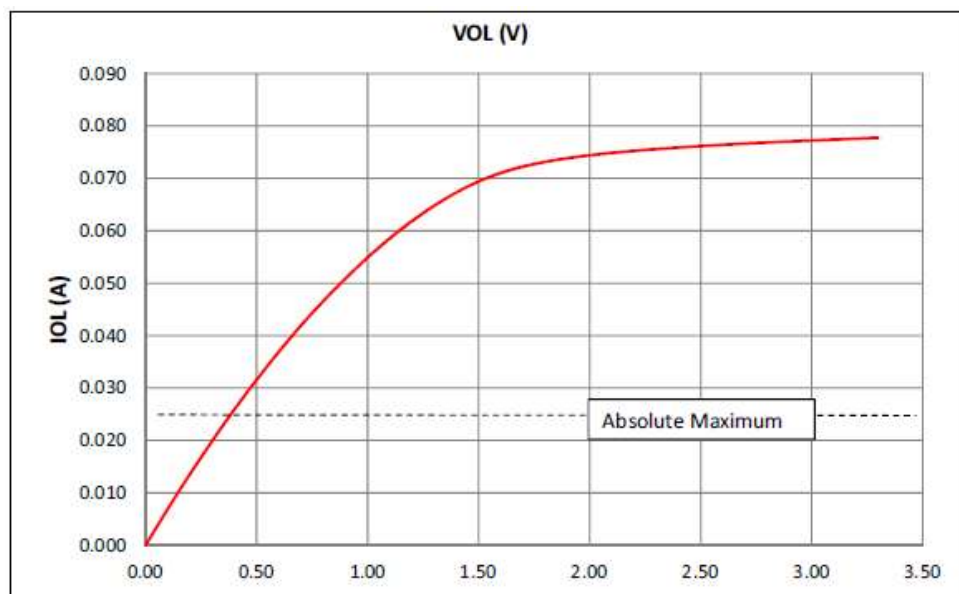
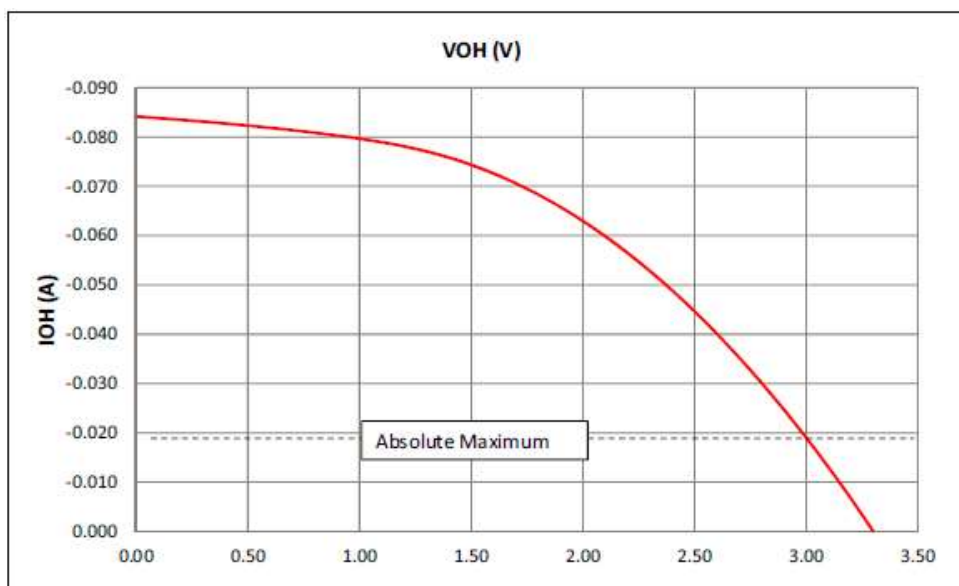


Figure 46-5. VOH vs IOH, I/O High Drive Strength I/O Pin Plot



46.13 Internal Voltage Reference Electrical Specifications (IVREF)

Table 46-18. Internal Voltage Reference DC Electrical Specifications

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ. ⁽¹⁾	Max.	Units	Conditions
VR_1	IVREF	ADC Internal Analog Channel Voltage	(*)	1.20326	(*)	V	ADC.CTRLB.ADCHSEL = 1.2V IVREF

Table 46-18. Internal Voltage Reference DC Electrical Specifications (continued)

DC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ. ⁽¹⁾	Max.	Units	Conditions
VR_9	ACIREF	Internal Comparator Voltage Reference	(*)	(*)	(*)	V	AC.COMPCTRLn.MUXNEG = Bandgap
VR_25	TDRIFT	Internal Voltage Reference Temperature Drift	—	—	(*)	%/°C	Over full operating temperature range
VR_27	VDRIFT	Internal Voltage Reference Voltage Drift	—	—	(*)	%/V	Over full operating voltage range

Note:

1. Typical values at 25°C, see parameter CMP_23 for Comparator Reference voltage range information.

46.14 Maximum Clock Frequencies

Table 46-19. Maximum Clock Frequencies AC Electrical Specifications

AC CHARACTERISTICS		Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial		
Param. No.	Symbol	Characteristics	Max	Units
FCLK_1	FCY	MCU clock freq	72	MHz
FCLK_3	fAHB	AHB clock freq	72	MHz
FCLK_5	fAPB0:1]	APB clock freq	72	MHz
FCLK_5	fAPB2		53	MHz
FCLK_6	fGCLKGEN[0]	GCLK clock freq output	200	MHz
	fGCLKGEN[1:7]		200	MHz
	fGCLKGEN[8:11]		100	MHz
FCLK_7	fGCLK_PLL	PLL reference clock freq	60	MHz
FCLK_11	fGCLK_DFL48M_REF	DFLL 48M reference clock freq	1	MHz
FCLK_13	fGCLK_EIC	EIC input clock freq	100	MHz
FCLK_15	fGCLK_FREQM_MSR	FREQM Measure Clock Frequency	200	MHz
FCLK_17	fGCLK_FREQM_REF	FREQM Reference Clock Frequency	100	MHz
FCLK_19	fGCLK_EVSYS_CHANNELx	EVSYS channel x input clock freq	100	MHz
FCLK_21	fGCLK_SERCOMx_SLOW	Common SERCOM slow input clock freq	32	kHz
FCLK_23	fGCLK_SERCOM_CORE	SERCOM input clock freq	160	MHz
FCLK_25	fGCLK_CANx	CAN input clock freq	100	MHz
FCLK_27	fGCLK_USBF5	USB FS input clock freq	48	MHz
FCLK_35	fGCLK_TCCx	TCCx input clock freq	200	MHz
FCLK_43	fGCLK_CCL	CCL input clock freq	50	MHz
FCLK_45	fGCLK_GCLKINx	External GCLKx input clock freq	50	MHz
FCLK_49	fGCLK_AC	Analog comparator peripheral module clock freq	100	MHz
FCLK_51	fGCLK_ADCx	ADCx input clock freq	144	MHz
FCLK_55	fGCLK_PTC	PTC input clock freq	50	MHz
FCLK_75	fGCLK_PUF	PUF input clock freq	80	MHz

Table 46-19. Maximum Clock Frequencies AC Electrical Specifications (continued)

AC CHARACTERISTICS		Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial		
Param. No.	Symbol	Characteristics	Max	Units
FCLK_77	fGCLK_AT	Anti-Tamper input clock freq	80	MHz

46.15 XOSC Electrical Characteristics

Table 46-20. External XTAL and Clock AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions ⁽¹⁾
XOSC_1	FOSC_XOSC	XOSC Crystal Frequency	4	—	48	MHz	XOSCCTRLn.XTALEN = 1 XIN, XOUT Primary Osc
XOSC_1A	TOSC	TOSC = 1 / FOSC_XOSC	20.83	—	250	ns	See parameter XOSC1 for FOSC_XOSC value
XOSC_2	XOSC_ST ⁽¹⁾	XOSC Crystal Start-up Time	—	—	1300000 ⁽³⁾	TOSC	Crystal stabilization time only, not Oscillator Ready XOSCCTRLA.AGC = 1 XOSCCTRLA.BW = 1 FOSC = 48 MHz (Note 1)
XOSC_3	CXIN	XOSC XIN parasitic pin capacitance	—	2.2	—	pF	—
XOSC_5	CXOUT	XOSC XOUT parasitic pin capacitance	—	2.2	—	pF	—

Table 46-20. External XTAL and Clock AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions ⁽¹⁾
XOSC_11	CLOAD ⁽²⁾	Crystal load capacitance FOSC_XOSC = 4 MHz	—	—	20	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 500Ω
XOSC_13		Crystal load capacitance FOSC_XOSC = 8 MHz	—	—		pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 500Ω
XOSC_15		Crystal load capacitance FOSC_XOSC = 12 MHz	—	—		pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 230Ω
XOSC_17		Crystal load capacitance FOSC_XOSC = 16 MHz	—	—	18	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 160Ω
XOSC_19		Crystal load capacitance FOSC_XOSC = 24 MHz	—	—		pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 70Ω
XOSC_21		Crystal load capacitance FOSC_XOSC = 32 MHz	—	—	12	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 80Ω
XOSC_23		Crystal load capacitance FOSC_XOSC = 48 MHz	—	—	8	pF	XOSCCTRLA.AGC = 1 Crystal ESR ≤ 70Ω
XOSC_33	DLEVEL	MCU Crystal Osc Power Drive Level	—	—	100	μW	XOSCCTRLA.AGC = 1
XOSC_35	FOSC_XCLK	Ext Clock Oscillator Input Freq (XIN pin)	4	—	48	MHz	XOSCCTRLA.XTALEN = 0
XOSC_37	XCLK_DC	Ext Clock Oscillator (XIN) Duty Cycle	40	50	60	%	XOSCCTRLA.XTALEN = 0
XOSC_39	XCLK_FST	Primary XIN Clock Fail Safe Time-out Period	—	$4 \times 1 / (DFLL\_1/2 \times XOSCCTRLA.CFDPRESC)$	—	μs	—

Table 46-20. External XTAL and Clock AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions ⁽¹⁾

Notes:

- This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern, the customer would need to characterize this based on their design choices.
- The Crystal Load Capacitor calculation is as follows:
 - Standard PCB trace capacitance = 1.5 pF per 12.5 mm (0.5 inches) (i.e., PCB STD TRACE W = 0.175 mm, H = 36 μm, T = 113 μm)
 - XTAL PCB capacitance typical therefore ≈ 2.5 pF for a tight PCB XTAL layout
 - For C_{XIN} and C_{XOUT} within 4 pF of each other, Assume C_{XTAL_EFF} = ((C_{XIN} + C_{XOUT}) / 2)

Note: Averaging C_{XIN} and C_{XOUT} will effect final calculated C_{LOAD} value by less than 0.25 pF.

Notes:

Equation 1:

MFG C_{LOAD} Spec = { ([C_{XIN} + C1] * [C_{XOUT} + C2]) / [C_{XIN} + C1 + C2 + C_{XOUT}] } + estimated oscillator PCB stray capacitance

- Assuming C1 = C2 and C_{XIN} ≈ C_{XOUT}, the formula can be further simplified and restated to solve for C1 and C2 by:

Equation 2 (Simplified version of equation 1):

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

For example:

- XTAL Mfg C_{LOAD} Data Sheet Spec = 12 pF
- PCB XTAL trace Capacitance = 2.5 pF
- C_{XIN} pin = 6.5 pF, C_{XOUT} pin = 4.5 pF, therefore C_{XTAL_EFF} = ((C_{XIN} + C_{XOUT}) / 2)

C_{XTAL_EFF} = ((6.5 + 4.5)/2) = 5.5 pF

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

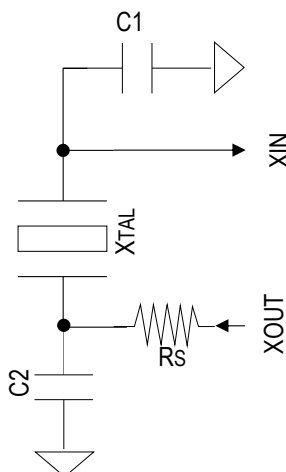
C1 = C2 = (24 - 5.5 - (2 * 2.5))

C1 = C2 = 13.5 pF (Always rounded down)

C1 = C2 = 13 pF (i.e., for hypothetical example crystal external load capacitors)

User C1 = C2 = 13 pF ≤ C_{LOAD}(max) spec

Figure 46-6. XTAL



- Maximum start up time; user must configure XOSCCTRLA.STARTUP counter value greater than start up time.

46.16 XOSC32K Electrical Characteristics

Table 46-21. XOSC32K AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
XOSC32_1	FOSC_XOSC32	XOSC32 Oscillator Crystal Frequency	—	32.768	—	kHz	XIN32, XOUT32 Secondary Osc
XOSC32_3	CXIN32	XOSC32 XIN32 parasitic pin capacitance	—	4.9	—	pF	—
XOSC32_5	CXOUT32	XOSC32 XOUT32 parasitic pin capacitance	—	4.8	—	pF	—
XOSC32_11	CLOAD_X32 ⁽²⁾	32.768 kHz Crystal Load Capacitance	—	—	1.5	pF	XOSC32K.CGM ≥ 6 XOSC32K.BOOST = 0 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1
XOSC32_13	ESR_X32	32.768 kHz Crystal ESR	—	—	60	KΩ	XOSC32K.CGM ≥ 6 XOSC32K.BOOST = 0 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1 Clod = 12.5 pF
XOSC32_14			—	—	100	KΩ	XOSC32K.CGM ≥ 11 XOSC32K.BOOST = 0 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1 Clod = 12.5 pF
XOSC32_15	TOSC32	TOSC32 = 1/ FOSC_XOSC32	—	30.5176	—	μs	See parameter XOSC32_1 for FOSC_XOSC32 value
XOSC32_17	XOSC32_ST ⁽¹⁾	XOSC32 Crystal Start-up Time	—	—	14000 ³	TOSC	XOSC32K.CGM ≥ 6 XOSC32K.BOOST = 0 XOSC32K.XTALEN = 1 XOSC32K.ENABLE = 1 Crystal ESR = 100 KΩ Clod = 12.5 pF ⁽¹⁾ Crystal stabilization time only not Oscillator Ready
XOSC32_19	FOSC_XCLK32	Ext Clock Oscillator Input Freq (XIN32 pin)	31.13	32.768	34.406	kHz	XOSC32K.XTALEN = 0 XOSC32K.ENABLE = 1

Table 46-21. XOSC32K AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
XOSC32_21	XCLK32_DC	Ext Clock Oscillator Duty Cycle	25	50	75	%	XOSC32K.XTALEN = 0 XOSC32K.ENABLE = 1
XOSC32_23	XCLK32_FST	XIN32 Clock Fail Safe Time-out Period	—	$4 \times 1 / (LP32K\_1/2^{CFDCTRL.CFDPRESC})$	—	μs	—

Table 46-21. XOSC32K AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

- This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern the customer would need to characterize this based on their design choices.
- The Crystal Load Capacitor calculation is as follows:
 - Standard PCB trace capacitance = 1.5 pF per 12.5 mm (0.5 inches) (i.e., PCB STD TRACE W = 0.175 mm, H = 36 μm, T = 113 μm)
 - XTAL PCB capacitance typical, therefore ~ = 2.5 pF for a tight PCB XTAL layout
 - For C_{XIN32} and C_{XOUT32} within 4 pF of each other, assume C_{XTAL_EFF} = ((C_{XIN32} + C_{XOUT32}) / 2)

Note: Averaging C_{XIN32} and C_{XOUT32} will effect final calculated C_{LOAD} value by less than 0.25 pF.

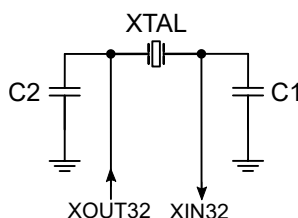
Equation 1:

MFG C_{LOAD} Spec = { ([C_{XIN32} + C1] * [C_{XOUT32} + C2]) / [C_{XIN32} + C1 + C2 + C_{XOUT32}] } + estimated oscillator PCB stray capacitance

- Assuming C1 = C2 and C_{XIN32} ~ = C_{XOUT32}, the formula can be further simplified and restated to solve for C1 and C2 by:

Equation 2 (Simplified Equation 1):

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))



For example:

- XTAL Mfg C_{LOAD} Data Sheet Spec = 12 pF
- PCB XTAL Trace Capacitance = 2.5 pF
- C_{XIN32} pin = 6.5 pF, C_{XOUT32} pin = 4.5 pF. Therefore, C_{XTAL_EFF} = ((C_{XIN32} + C_{XOUT32}) / 2)

C_{XTAL_EFF} = ((6.5 + 4.5)/2) = 5.5 pF

C1 = C2 = ((2 * MFG C_{LOAD} spec) - C_{XTAL_EFF} - (2 * PCB capacitance))

C1 = C2 = (24 - 5.5 - (2 * 2.5))

C1 = C2 = (24 - 5.5 - 5)

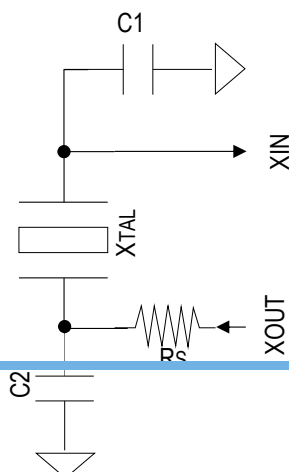
C1 = C2 = 13.5 pF (always rounded down)

C1 = C2 = 13 pF (i.e., for hypothetical example crystal external load capacitors)

User C1 = C2 = 13 pF ≤ C_{LOAD_X32}(max) spec

- User Selectable in XOSC32K.STARTUP.

Figure 46-7. XTAL



46.17 OSCULP32K Electrical Characteristics

Table 46-22. Low-Power Internal 32 kHz RC Oscillator AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
LP32K_1	FOSC_LPRC32K	Output Frequency	(*)	32.768	(*)	kHz	-40°C ≤ TA ≤ +85°C (i.e., Factory Default Calibration)
LP32K_9	RC32K_Duty	LPRC32K OSC Duty Cycle	—	50	—	%	

46.18 DFLL-FDPLL Electrical Characteristics

Table 46-23. DFLL AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DFLL48 MHz (Open Loop) ^(1,2)							
DFLL_1	DFLL_OL_FOUT	DFLL Open Loop Clock Frequency	45.600	48.000	50.400	MHz	Normal mode (DFLLCTRLA.LOWFREQ = 0)
			7.037	8.000	8.962		Low-Frequency mode (DFLLCTRLA.LOWFREQ = 1)
DFLL_9	DFLL_OL_SRT	Start-Up (Ready bit valid)	—	13	16	μs	Normal mode (DFLLCTRLA.LOWFREQ = 0)
			—	68	75		Low-Frequency mode (DFLLCTRLA.LOWFREQ=1)
DFLL48 MHz (Closed Loop) ^(3,4)							
DFLL_11	DFLL_CL_FIN	DFLL Closed Loop Input Frequency Range ⁽⁴⁾	1000	32768	1000000	Hz	—
DFLL_13	DFLL_CL_FOUT	DFLL Closed Loop Clock Frequency ⁽⁶⁾	47.88	48.00	48.12	MHz	Normal mode (DFLLCTRLA.LOWFREQ = 0) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 1465
DFLL_14			7.98	8.00	8.02		Low-Frequency mode (DFLLCTRLA.LOWFREQ = 1) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 244
DFLL_15	DFLL_CL_Jitter	DFLL Period Jitter Pk-to-Pk	—	0.920	2.700	%	Normal mode (DFLLCTRLA.LOWFREQ = 0) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 1465
DFLL_17			—	1.000	2.000	%	Low- Frequency mode (DFLLCTRLA.LOWFREQ = 1) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 244

Table 46-23. DFLL AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DFLL_21	DFLL_CL_SRT	DFLL Closed- Loop mode / Lock Time ^(5, 6)	—	0.460	1.100	ms	Normal mode (DFLLCTRLA.LOWFREQ = 0) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 1465
DFLL_23			—	0.350	1.000		Low-Frequency mode (DFLLCTRLA.LOWFREQ = 1) XOSC32 32.768 kHz PPM ≤ 100, DFLLMUL = 244

Table 46-24. PLL (Frequency Digital Phase Locked Loop) AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
PLLxxMHz (Digital Phase-Locked Loop)							
PLL_1	PLL_FIN	PLL Input Frequency Range	4	—	48	MHz	—
PLL_3	PLL_FOUT	PLL Output Clock Frequency	12.7	—	200	MHz	
PLL_11	PLL_SRT	Lock Time ⁽⁷⁾	—	25	—	μs	—
Notes: - In Open-Loop mode, the DFLL uses a self contained internal RC oscillator clock source. its course calibrated value is loaded at out of reset. In addition, there is a fine tune trim register (DFLLTUNE) the user software can assess. - Not recommended for functional USB operation, SOF sync start-up only. - In Closed-Loop mode the DFLL can use a variety of clock sources. The DFLL can be trimmed using the DFLLMUL register. - To ensure that the DFLL stays within the +/-0.25% of its clock frequency, any reference clock for DFLL in close loop must be within a 8% maximum error accuracy. - REFCLK for DFLL or XOSC32K, PLL is XOSC. - DFLLCTRLB.QLDIS = 0: quick lock enabled (default), DFLLCTRLB.CCDIS = 0: chill cycles enabled (default). DFLLMUL.STEP = 8: Maximum fine step size, divided or dividing into two parts, search 8 is optimum value. During a maximum 30 cycles of the reference clock period, between lock flag asserted and frequency stabilization, DFLL accuracy will be limited to +/-1.5%. After frequency stabilization has been achieved, the accuracy will be +/-0.25%. Disabling Quick Lock and reducing the STEP value at 4 (instead of the optimum 8) will eliminate this period of inaccuracy.							

46.19 Analog-to-Digital Converter (ADC) Electrical Specifications

Table 46-25. ADC AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
Device Supply							
ADC_1	AVDD	ADC Module Supply	AVDD(min)	—	AVDD(max)	V	VDDIOx = AVDD
Reference Inputs							
ADC_3	VREF ⁽¹⁾	ADC Reference Voltage	2.4	—	AVDD	V	VREF ≤ AVDD
Analog Input Range							
ADC_7	AFS	Full-Scale Analog Input Signal Range	AVSS	—	VREF	V	Single-Ended mode
			-VREF	—	+VREF	V	Differential mode, VCMIN = VREF/2
ADC_9	VCMIN	Input common mode voltage	—	VREF/2	—	V	—
Note:							
1. ADC functional device operation with either internal or external VREF<2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately $\sim((0.06 * 2n) / VREF)$ LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on AVDD, AVSS.							

Table 46-26. ADC Single Ended Mode AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
Single Ended Mode ADC Accuracy⁽⁴⁾							
SADC_11	Res	Resolution	8	—	12	bits	Selectable 8, 10, 12 bit Resolution Ranges
SADC_13	INL ⁽³⁾	Integral Nonlinearity	-2.2	-1.7/1.1	1.3	LSB	4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V
SADC_19	DNL ⁽³⁾	Differential Nonlinearity	-1.4	-1/1.8	2.4	LSB	4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V
SADC_25	GERR ^(3,6)	Gain Error	-4.7	—	-1.2	LSB	4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V
SADC_31	EOFF ^(3,6)	Offset Error	1.9	—	5.1	LSB	4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V
Single Ended Mode ADC Dynamic Performance^(1,2,4)							

Table 46-26. ADC Single Ended Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SADC_43	ENOB ⁽³⁾	Effective Number of bits	10.3	10.75	—	bits	VREF = AVDD = VDDIO = 3.3V at 12bit at 4.25 Msps
SADC_45	SINAD ^(1,2,3)	Signal-to-Noise and Distortion	63	66	—	dB	
SADC_47	SNR ^(1,2,3)	Signal-to-Noise ratio	64	66	—		
SADC_51	THD ^(1,2,3,5)	Total Harmonic Distortion	—	-81	-77		

Notes:

1. Characterized with an analog input sine wave = (FTP (max) / 100). For example, FTP(max) = 1 Msps / 100 = 10 kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12-bit resolution.
3. ADC is configured in 12 bits mode, All registers are at the reset default value unless otherwise stated.
4. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately $\sim((0.06 * 2n) / VREF)$ LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on VDDANA, GNDANA.
5. Value taken over 7 harmonics.
6. Value coming from simulation.

Table 46-27. ADC Differential Mode AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
Differential Mode ADC Accuracy ⁽⁴⁾							
DADC_11	Res	Resolution	8	—	12	bits	Selectable 8, 10, 12 bit Resolution Ranges
DADC_13	INL ⁽³⁾	Integral Nonlinearity	-2.3	-1.6/1.5	2.2	LSB	4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V
DADC_19	DNL ⁽³⁾	Differential Nonlinearity	-1	-1/1.9	2.7	LSB	4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V
DADC_25	GERR ^(3,6)	Gain Error	-4.3	—	0	LSB	4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V
DADC_31	EOFF ^(3,6)	Offset Error	-0.1	—	1.9	LSB	4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V
Differential Mode ADC Dynamic Performance ^(1,2,4)							
DADC_43	ENOB ⁽³⁾	Effective Number of bits	10.5	11.25	—	bits	VREF = AVDD = VDDIO = 3.3V at 12 bit at 4.5 Msps
DADC_45	SINAD ^(1, 2, 3)	Signal-to-Noise and Distortion	65	69	—	dB	
DADC_47	SNR ^(1, 2, 3)	Signal-to-Noise ratio	65	69	—		
DADC_51	THD ^(1, 2, 3, 5)	Total Harmonic Distortion	—	-86	-80		

Table 46-27. ADC Differential Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions

Notes:

1. Characterized with an analog input sine wave = (FTP (max) / 100). For example, FTP (max) = 1 Msps / 100 = 10 kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12-bit resolution.
3. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately $\sim(0.06 * 2^n) / VREF$ LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on VDDANA, GNDANA.
4. Value taken over 7 harmonics.
5. Value coming from simulation.

Table 46-28. ADC Conversion and Sample AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC Clock Requirements							
ADC_53	TAD	ADC Clock Period	13.88	—	7145	ns	VREF = AVDD = 3.3V and Res = 6,8,10 bit
			13.88	—	1250	ns	VREF = AVDD = 3.3V and Res = 12 bit
ADC_55	fGCLK_ADCx	ADCx Module GCLK max input freq	—	—	FCLK_51	MHz	VREF = AVDD = 3.3V
ADC Throughput Rates							
ADC_57	FTPR ⁽¹⁾	Sample-Rate for ADC0 with SAMC = 0 (min)	—	—	4.5	Msps	12-bit resolution, Rsource Impedance ≤ 200 Ω
			—	—	5.14		10-bit resolution, Rsource Impedance ≤ 350 Ω
			—	—	6		8-bit resolution, Rsource Impedance ≤ 500 Ω
			—	—	7.2		6-bit resolution, Rsource Impedance ≤ 650 Ω
ADC Conversion and Sample time							

Table 46-28. ADC Conversion and Sample AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
ADC_59	TSAMP	Sample-Time for ADC0	3	—	—	TAD	12 bit TAD(min), Ext Analog Input Rsource ≤ 200 Ω, Max ADC Clock
							10 bit TAD(min), Ext Analog Input Rsource ≤ 350 Ω, Max ADC Clock
			8	—	—		12 bit TAD(min), Ext Analog Input Rsource ≤ 500 Ω, Max ADC Clock
							10 bit TAD(min), Ext Analog Input Rsource ≤ 700 Ω, Max ADC Clock
			14	—	—		12 bit TAD(min), Ext Analog Input Rsource ≤ 1 kΩ, Max ADC Clock
							10 bit TAD(min), Ext Analog Input Rsource ≤ 1.25 kΩ, Max ADC Clock
			64	—	—		12 bit TAD(min), Ext Analog Input Rsource ≤ 5 kΩ, Max ADC Clock
							10 bit TAD(min), Ext Analog Input Rsource ≤ 5.5 kΩ, Max ADC Clock
ADC_61	TCNV	Conversion Time (after sample time is complete)	13			TAD	12-bit resolution
			11				10-bit resolution
			9				8-bit resolution
			7				6-bit resolution
ADC_63	Twarm-up	Warm Up Time after CTRLA.ANAEN = 1 and CTRLA.ENABLE = 1	—	—	500 TAD or 20 μs, which ever is bigger	μs	—

Notes:

- ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)).
- Specification values assume only one AINx channel in use.

46.20 Comparator (AC) Electrical Characteristics

Table 46-29. Comparator AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ. ⁽¹⁾	Max.	Units	Conditions
CMP_1	VIOFF	Input Offset Voltage	-8	—	8	mV	COMPCTRLn.HYST = 0x3. Comparator ref voltage = AVDD/2
CMP_4	VIN	Input Voltage Range	AVSS	—	AVDD	V	With respect to GND and AVDD

Table 46-29. Comparator AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ. ⁽¹⁾	Max.	Units	Conditions
CMP_5	VHYST_00	Input Hysteresis Voltage	2	—	14	mV	COMPCTRLn.HYST = 0x0. Comparator ref voltage = AVDD/2
	VHYST_01	Input Hysteresis Voltage	6	—	26	mV	COMPCTRLn.HYST = 0x1. Comparator ref voltage = AVDD/2
	VHYST_02	Input Hysteresis Voltage	20	—	56	mV	COMPCTRLn.HYST = 0x2. Comparator ref voltage = AVDD/2
	VHYST_03	Input Hysteresis Voltage	32	—	86	mV	COMPCTRLn.HYST = 0x3. Comparator ref voltage = AVDD/2
CMP_15	TRESPSS ain<0:1>	Small Signal Response Time - High speed	—	—	126	ns	COMPCTRLn.SPEED = 0x0 COMPCTRLn.MUXPOS = 0x0 COMPCTRLn.MUXNEG = 0x1 COMPCTRLn.HYST = 0x3 Comparator ref voltage = AVDD/2. Input overdrive = +/- 180 mV
		Small Signal Response Time - Low Speed	—	—	432	ns	COMPCTRLn.SPEED = 0x1 COMPCTRLn.MUXPOS = 0x0 COMPCTRLn.MUXNEG = 0x1 COMPCTRLn.HYST = 0x3 Comparator ref voltage = AVDD/2. Input overdrive = +/- 180 mV
CMP_19	COUTVAL	Comparator Enabled to Output Valid - High Speed	—	—	2	ms	Comparator module is configured before enabling it COMPCTRLn.SPEED = 0x0
		Comparator Enabled to Output Valid - Low Speed	—	—	20	ms	Comparator module is configured before enabling it COMPCTRLn.SPEED = 0x1
CMP_23	CVREFRNG	Comparator Voltage Reference Input Range	—	—	—	V	—
CMP_25	FGCLK_AC	Analog comparator peripheral module clock freq	—	—	FCLK_49	MHz	See FCLK_49 in Maximum Clock Frequencies

Notes:

- Typical values at 25°C.
- Comparator Ref voltage cannot exceed: $VIN(max) - VIOFF(max) - CMP\_5(max) - 100\text{ mV} \geq CMP\_23 \geq (VIN(min) + abs(VIOFF(min))) + CMP\_5(max) + 100\text{ mV}$.

Table 46-30. DAC Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DAC_1	DRES	DAC Resolution	—	—	7	Bits	—
DAC_7	VOUT	Output Voltage Range	AVSS	—	AVDD - 1LSB	V	—

Table 46-30. DAC Module AC Electrical Specifications (continued)

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
DAC_17	Tset_FS	DAC Full Scale Settling Time	—	—	1.5	μs	+/-4 LSB of final value for step size from 10% to 90%
DAC_18a	Thold	DAC Holding Time	0.29	—	—	ms	This parameter applies only for Low-Power mode: DACCTRLn.SHENm = 0x1 Time for 1 LSB voltage loss of the DAC output
DAC_18b			0.65	—	—	ms	This parameter applies only for Low-Power mode: DACCTRLn.SHENm = 0x1 Time for 10 LSB voltage loss of the DAC output
SDAC_19	INL	Integral Non Linearity	-0.4	—	0.4	LSB	—
SDAC_21	DNL	Differential Non Linearity	-0.5	—	0.4	LSB	—
SDAC_23	GERR	Gain Error	-1.2	—	-0.2	LSB	—
SDAC_25	EOFF	Offset Error	-0.7	—	-0.7	LSB	Normal mode: DACCTRLn.SHENm = 0x0
			-5.7	—	5.7	LSB	Low-Power mode: DACCTRLn.SHENm = 0x1

46.21 Peripheral Touch Controller (PTC) Electrical Characteristics

Table 46-31. Peripheral Touch Controller (PTC) AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
PTC_1	CLOAD SC ⁽¹⁾	Self Capacitance mode (PTC Channel Y0 - Y35)	Y0	—	—	65	—
			Y1-35	—	—	70	pF
PTC_3	CLOAD MC ⁽¹⁾	Mutual Capacitance mode (PTC Channel Y0 - Y35)	—	—	35	pF	Maximum sensor load capacitance ⁽¹⁾
PTC_4A	fPTC ⁽²⁾	PTC frequency	—	—	4	MHz	—
PTC_4B	Twarm	Warm-up time	—	—	500 PTC clock cycles	μs	—

Notes:

- Maximum capacitive load that the PTC circuitry can compensate for each channel.
- fPTC = GCLK_PTC / PTC prescaler.

46.22 Serial Peripheral Interface (SPI) Timings Electrical Characteristics

Figure 46-8. SPIx Host Module (CPHA = 0) Timing Diagrams

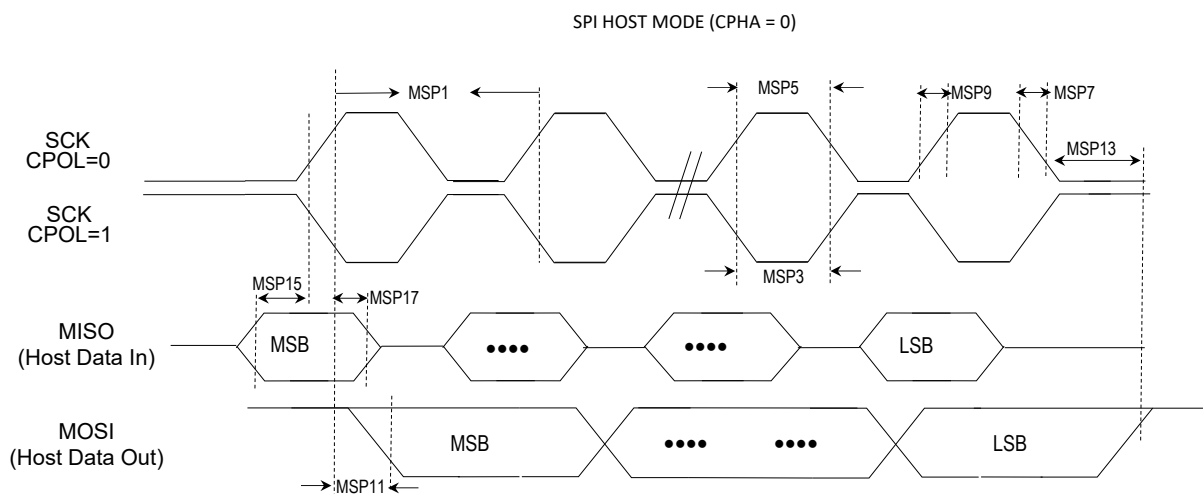


Figure 46-9. SPIx Host Module (CPHA = 1) Timing Diagrams

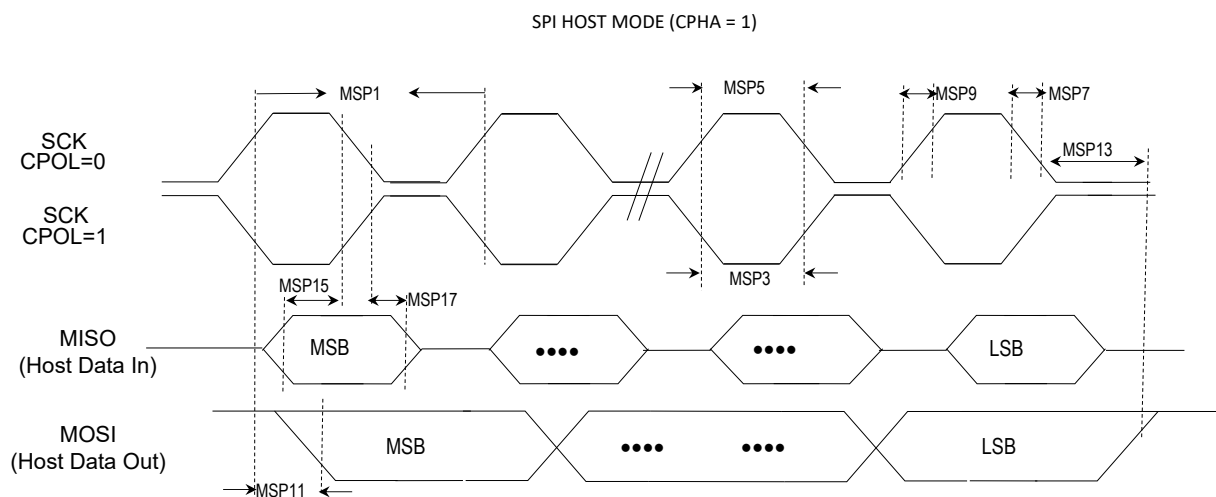


Table 46-32. SPIx Module Host Mode Electrical Specifications ⁽¹⁾

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Typ.	Max.	Units	Conditions
MSP_1	FSCK	SCK Frequency	Normal Speed SERCOM	—	—	22	MHz	Transmitter mode, CTRB.RXEN = 0. CLOAD = 30 pF(max)
				—	—	22		Full Duplex Transmit and Receive mode, Loop-Back mode with one SPI talking to another SPI on the same MCU. CLOAD = 30 pF(max)
				—	—	1/(2*(TMIS+NOTE2_TV)) ⁽²⁾		Full Duplex Transmit and Receive mode, CLOAD = 30 pF (max). The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop-Back mode above may therefore be possible using the formula.
		High-Speed SERCOM ⁽³⁾	—	—	46	MHz	Transmitter mode, CTRB.RXEN = 0, VDDIOx >= 2.97V, CLOAD = 20 pF(max)	
			—	—	1/(2*(TMIS_HS+NOTE2_TV)) ⁽²⁾		Full-Duplex Transmit and Receive mode, CLOAD=20 pF(max). The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop- Back mode above may therefore be possible using the formula. VDDIOx >= 2.97V	
MSP_3	TSCL	SCK Output Low Time		1/(2*FSCK)	—	—	ns	—
MSP_5	TSCH	SCK Output High Time		1/(2*FSCK)	—	—	ns	—
MSP_7	TSCF	SCK and MOSI Output Fall Time		—	—	DI_27 or DI_31	ns	See parameter in I/O Specifications
MSP_9	TSCR	SCK and MOSI Output Rise Time		—	—	DI_25 or DI_29	ns	See parameter in I/O Specifications

Table 46-32. SPIx Module Host Mode Electrical Specifications ⁽¹⁾ (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Typ.	Max.	Units	Conditions
MSP_11	TMOV	MOSI Data Output Valid after SCK	Normal Speed SERCOM	—	—	9.56	ns	—
			High-Speed SERCOM ⁽³⁾	—	—	2.94	ns	VDDIOx ≥ 2.97V
MSP_13	TMOH	MOSI hold after SCK		0	—	—	ns	—
MSP_15	TMIS	MISO Setup Time of Data Input to SCK	Normal Speed SERCOM	12.8	—	—	ns	—
			High-Speed SERCOM ⁽³⁾	1.31	—	—	ns	VDDIOx ≥ 2.97V
MSP_17	TMIH	MISO Hold Time of Data Input to SCK	Normal Speed SERCOM	0	—	—	ns	—
			High Speed SERCOM ⁽³⁾	1.49	—	—	ns	VDDIOx ≥ 2.97V
MSP_19	SPI_GCLK	SERCOM SPI input clk freq, GCLK_SPI		—	—	FCLK_23	MHz	—

Notes:

- Assumes 30 pF external load on all SPIx pins unless otherwise noted.
- NOTE2_TV is the client external device data output valid time from clock edge specification.
- SERCOM0 pins PC0,PC1,PC2,PC3.

Figure 46-10. SPIx Client Module (CPHA = 0) Timing Diagrams

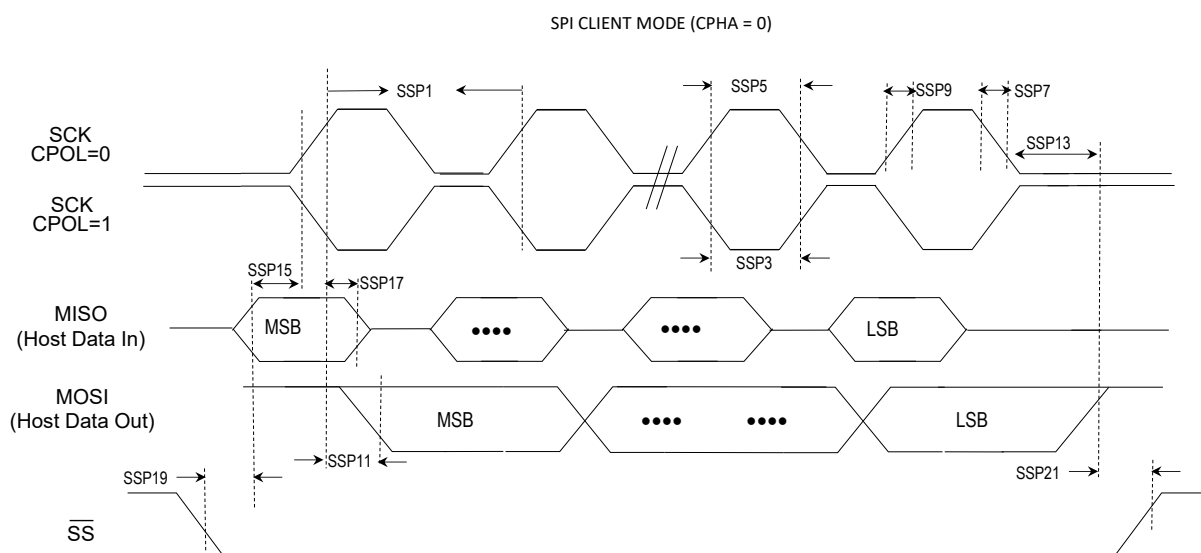


Figure 46-11. SPIx Client Module (CPHA = 1) Timing Diagrams

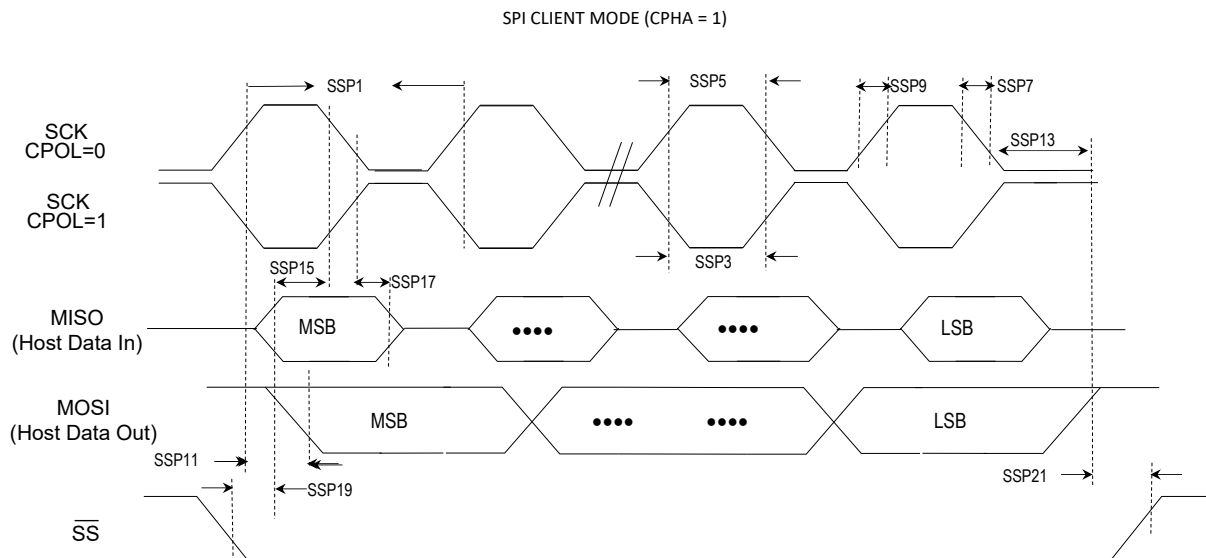


Table 46-33. SPIx Module Client Mode Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Typical	Max.	Units	Conditions
SSP_1	FCK	SCK Frequency	Normal Speed SERCOM	—	—	22	MHz	Receiver mode. CLOAD = 30 pF(max)
				—	—	22		Full Duplex Transmit and Receive mode, Loop-Back mode with one SPI talking to another SPI on the same MCU. CLOAD = 30 pF(max)
				—	—	1/(2*(TSOV+NOTE2_TMIS)) ⁽²⁾		Full Duplex Transmit and Receive mode, CLOAD=30 pF(max). The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the Loop-Back mode above may therefore be possible using the formula.
		SCK Frequency	High-Speed SERCOM ⁽³⁾	—	—	46	MHz	Receiver mode. VDDIOx ≥ 2.97V, CLOAD = 20 pF(max)
				—	—	1/(2*(TSOV_HS+NOTE2_TMIS)) ⁽²⁾		Full Duplex Transmit and Receive mode, CLOAD=20 pF(max). The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics. Faster speeds than the loop back mode above may therefore be possible using the formula. VDDIOx≥2.97V
				—	—	—		—
SSP_3	TSCL	SCK Output Low Time		1/(2*FCK)	—	—	ns	—
SSP_5	TSCH	SCK Output High Time		1/(2*FCK)	—	—	ns	—
SSP_7	TSCF	SCK and MOSI Output Fall Time		—	—	DI_27 or DI_31	ns	See parameter in I/O Specifications
SSP_9	TSCR	SCK and MOSI Output Rise Time		—	—	DI_25 or DI_29	ns	See parameter in I/O Specifications

Table 46-33. SPIx Module Client Mode Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Typical	Max.	Units	Conditions
SSP_11A	TSOV	MISO Data Output Valid after SCK	Normal-Speed SERCOM	—	—	14.83	ns	—
			High-Speed SERCOM ⁽³⁾	—	—	10.52	ns	VDDIOx ≥ 2.97V
SSP_13	TSOH	MISO Hold after SCK		4	—	—	ns	—
SSP_15	TSIS	MOSI Setup Time of Data Input to SCK	Normal-Speed SERCOM	7.33	—	—	ns	—
			High-Speed SERCOM ⁽³⁾	2.22	—	—	ns	VDDIOx ≥ 2.97V
SSP_17	TSIH	MOSI Hold Time of Data Input to SCK	Normal-Speed SERCOM	2.24	—	—	ns	—
			High-Speed SERCOM ⁽³⁾	1.92	—	—	ns	VDDIOx ≥ 2.97V
SSP_19	TSSS	$\overline{SS}$ setup to SCK (PRELOADEN = 0)	Normal-Speed SERCOM	5.76	—	—	ns	—
			High-Speed SERCOM ⁽³⁾	1.47	—	—	ns	VDDIOx ≥ 2.97V
SSP_21	TSSH	$\overline{SS}$ hold after SCK Client	Normal-Speed SERCOM	0	—	—	ns	—
			High-Speed SERCOM ⁽³⁾	0.36	—	—	ns	VDDIOx ≥ 2.97V
SSP_23	SPI_GCLK	SERCOM SPI input clk freq, GCLK_SPI	—	—	—	FCLK_23	MHz	—
Notes: 1. Assumes 30 pF external load on all SPIx pins unless otherwise noted. 2. NOTE2_TMIS is the host external device setup time. 3. SERCOM0 pins PC0,PC1,PC2,PC3.								

46.23 Universal Asynchronous Receiver/Transmitter (UART) Electrical Characteristics

Table 46-34. UART AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial						
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions		
UT_1A	FBRATE	Baud Rate - Normal Speed SERCOM	Asynchronous SAMPR = 16x mode	—	—	10	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
				—	—	10	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_3A			Asynchronous SAMPR = 8x mode	—	—	20	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
				—	—	20	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_5A			Asynchronous SAMPR = 3x mode	—	—	21	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
				—	—	20	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_19A			Synchronous Mode X2 mode	—	—	18	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
				—	—	12	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_21A			Synchronous Mode X1 mode	—	—	18	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
				—	—	12	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_1B			FBRATE_HS	Baud Rate - High Speed SERCOM	Asynchronous SAMPR = 16x mode	—	—	10	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)
						—	—	10	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)
UT_3B	Asynchronous SAMPR = 8x mode	—			—	20	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
		—			—	20	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_5B	Asynchronous SAMPR = 3x mode	—			—	21	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
		—			—	20	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_19B	Synchronous Mode X2 mode	—			—	21	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
		—			—	21	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_21B	Synchronous Mode X1 mode	—			—	21	Mbps	VDDIO = 3.3V, CLOAD = 30 pF(max)		
		—			—	21	Mbps	VDDIO = 1.8V, CLOAD = 30 pF(max)		
UT_23	FUSART	USART max GCLK_SERCOM			—	—	FCLK_23	MHz	—	
UT_25	FXCK	USART External Clock Input			—	—	25	MHz	—	

46.24 I²C Electrical Characteristics

Figure 46-12. I²C Start/Stop Bits Host Mode AC Timing Diagrams

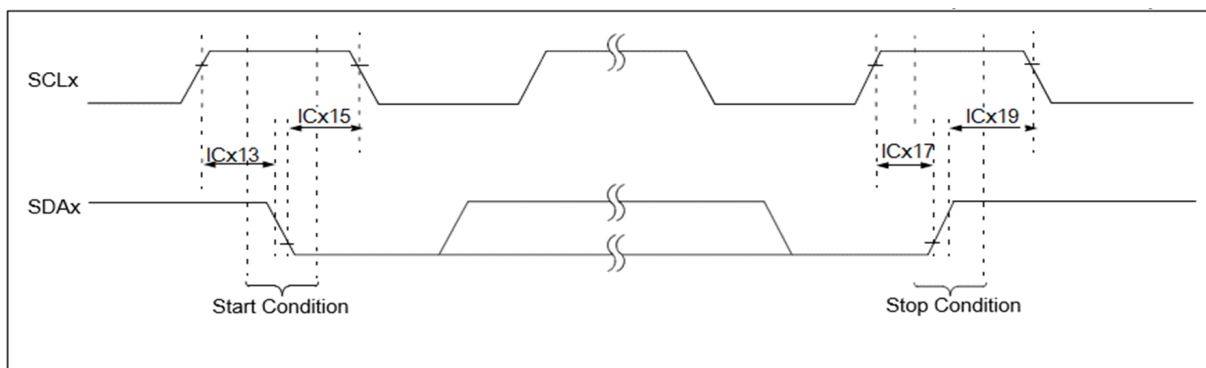


Figure 46-13. I²C Bus Data Host Mode AC Timing Diagrams

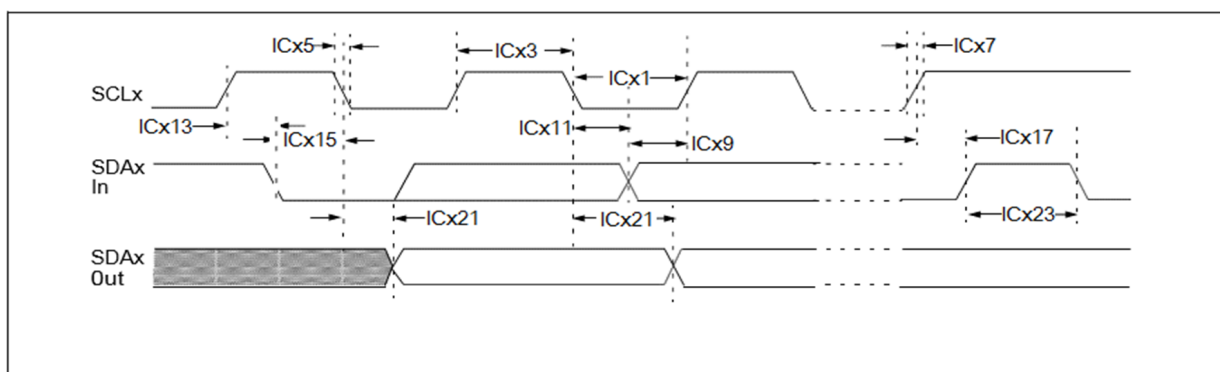


Table 46-35. I²C Host Mode AC Electrical Specifications

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics(1)		Min.	Max.	Units	Conditions
I2CM_1	TL0:SCL	Host Clock Low Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	μs	
			1 MHz mode	0.5	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 46-35. I²C Host Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics(1)		Min.	Max.	Units	Conditions
I2CM_3	THI:SCL	Host Clock High Time	100 kHz mode	4	—	μs	VDDIOx = 3.3V, IPULL-UP = 3 mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	60	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_5	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	VDDIOx = 3.3V, IPULL-UP = 3 mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_7	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_9	TSU:DAT	Data Setup Time	100 kHz mode	250	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	10	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 46-35. I²C Host Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics(1)		Min.	Max.	Units	Conditions
I2CM_11	THD:DAT ⁽¹⁾	Data Hold Time	100 kHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	300	—	ns	
			1 MHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	5	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_13	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_15	THD:STA	Start Condition Hold Time	100 kHz mode	4	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_17	TSU:ST0	Stop Condition Setup Time	100 kHz mode	4	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 46-35. I²C Host Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics(1)		Min.	Max.	Units	Conditions
I2CM_21	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3.45	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	0.9	μs	
			1 MHz mode	—	0.45	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	100	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CM_23	TBF:SDA ⁽²⁾	Bus Free Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	μs	
			1 MHz mode	0.5	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Notes:

- Longest delay between data hold timing based on bitfield SDAHOLD of register CTRLA from SERCOM Module and timing based on 4 period of GCLK_SERCOM for 100 kHz/400 kHz/1 MHz mode.
- The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

Figure 46-14. I²C Start/Stop Bits Client Mode AC Timing Diagrams

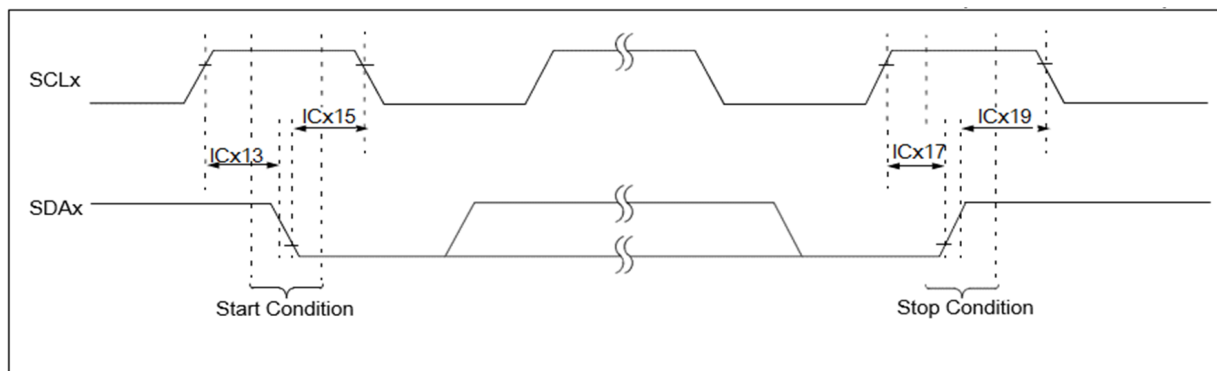


Figure 46-15. I²C Bus Data Client Mode AC Timing Diagrams

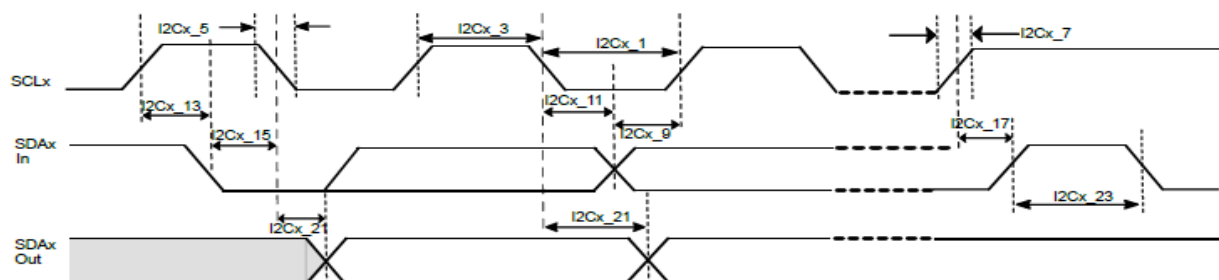


Table 46-36. I²C Client Mode AC Electrical Specifications

AC CHARACTERISTICS_				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Max.	Units	Conditions
I2CS_1	TL0:SCL	Client Clock Low Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	μs	
			1 MHz mode	0.5	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_3	THI:SCL	Client Clock High Time	100 kHz mode	4	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	60	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_5	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 46-36. I²C Client Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS_				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Max.	Units	Conditions
I2CS_7	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	300	ns	
			1 MHz mode	—	120	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	40	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_9	TSU:DAT	Data Setup Time	100 kHz mode	250	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	100	—	ns	
			1 MHz mode	50	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	10	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_11	THD:DAT ⁽¹⁾	Data Hold Time	100 kHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	300	—	ns	
			1 MHz mode	300	—	ns	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	5	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_13	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Table 46-36. I²C Client Mode AC Electrical Specifications (continued)

AC CHARACTERISTICS_				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial			
Param. No.	Symbol	Characteristics ⁽¹⁾		Min.	Max.	Units	Conditions
I2CS_15	THD:STA	Start Condition Hold Time	100 kHz mode	4	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_17	TSU:ST0	Stop Condition Setup Time	100 kHz mode	4	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	0.6	—	μs	
			1 MHz mode	0.26	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_21	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3.45	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	—	0.9	μs	
			1 MHz mode	—	0.45	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	—	100	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF
I2CS_23	TBF:SDA ⁽²⁾	Bus Free Time	100 kHz mode	4.7	—	μs	VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF
			400 kHz mode	1.3	—	μs	
			1 MHz mode	0.5	—	μs	VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF
			3.4 MHz mode	160	—	ns	VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF

Notes:

- Longest delay between data hold timing based on bitfield SDAHOLD of register CTRLA from SERCOM Module and timing based on 4 period of GCLK_SERCOM for 100 kHz/400 kHz/1 MHz mode.
- The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

46.25 Controller Area Network (CAN) Electrical Characteristics

Figure 46-16. CANx Module AC Timing Diagram

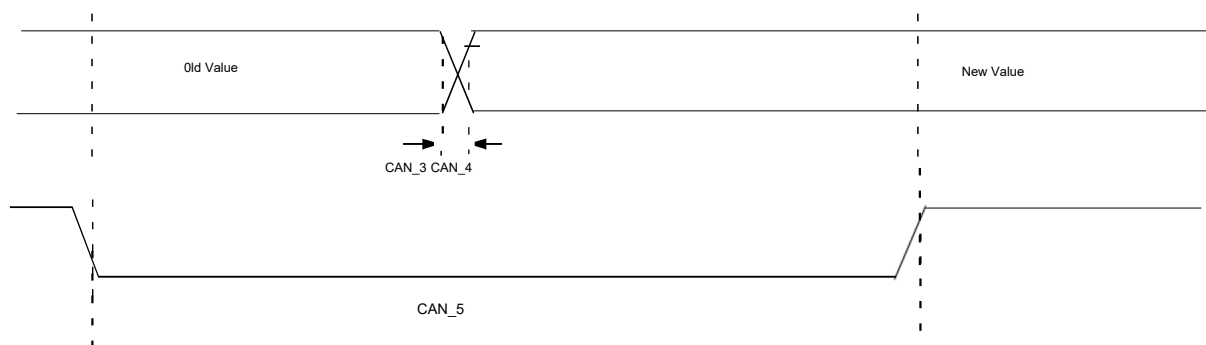


Table 46-37. CANx Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
CAN_1	CANRATE	CAN data rate	—	—	10	Mbps	VDDIOx = 3.3V, CLOAD = 30 pF(max)
			—	—	10		VDDIOx = 1.8V or VDDIO(min) whichever is greater, CLOAD = 30 pF(max)
CAN_3	CANFALL	Port Output Fall Time	—	—	DI_27 or DI_31	ns	See parameter in I/O Specifications
CAN_4	CANRISE	Port Output Rise Time	—	—	DI_25 or DI_29	ns	See parameter in I/O Specifications
CAN_7	fCAN_GCLK	CANx Input clk freq, GCLK_CAN	—	—	FCLK_25	MHz	VDDIOx = 1.8V

Note:

- Assumes VDDIOx(min) and 30 pF external load on all CAN pins unless otherwise noted.

46.26 Timer/Counter Controller (TCC) Electrical Characteristics

Figure 46-17. TCCx Timer Capture Input Module AC Timing Diagrams

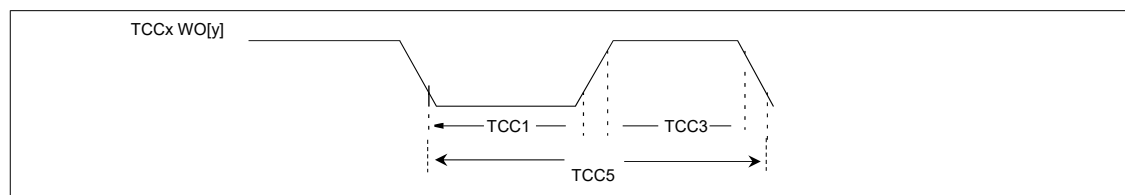


Figure 46-18. TCCx Timer Compare Output Module AC Timing Diagrams

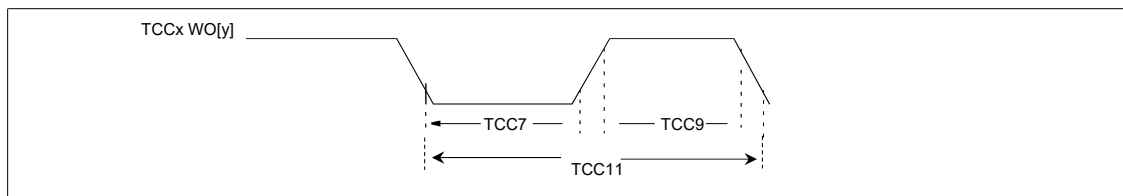


Figure 46-19. TCCx Timer Compare Fault Output Module AC Timing Diagrams

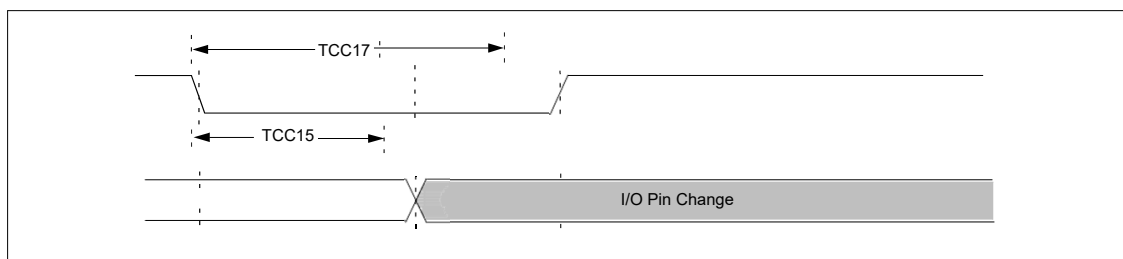


Table 46-38. TCCx Timer Capture Module AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
TCC_1	TCCINLOW	Capture TCCx Input Low Time	2/fGLK_TCCx	—	—	ns	VDDIOx(min) and meet TCC_5 spec
TCC_3	TCCINHIGH	Capture TCCx Input High Time	2/fGLK_TCCx	—	—	ns	VDDIOx(min) and meet TCC_5 spec
TCC_5	TCCINPERIOD	Capture Input Period	4/fGLK_TCCx	—	—	ns	VDDIOx(min)
TCC_7	TCCOUTLOW	Compare TCCx Output Low Time	3*DI_27 or 3*DI_31	—	—	ns	VDDIOx(min) and meet TCC_11 spec
TCC_9	TCCOUTHIGH	Compare TCCx Output High Time	3*DI_25 or 3*DI_29	—	—	ns	VDDIOx(min) and meet TCC_11 spec
TCC_11	TCCOUTPERIOD	Compare Output Period	TCC_7+TCC_9	—	—	ns	VDDIOx(min)
TCC_13	fGCLK_TCCx	TCC peripheral module clock frequency	—	—	FCLK_35	MHz	
TCC_15	TCCFD	Fault Input to I/O Pin Change	—	—	43	ns	
TCC_17	TCCFLT	Fault Input Pulse Width	1/2*(t_rise+t_fall)	—	—	ns	
Note:							
1. Assumes VDDIOx(min) and 20 pF external load on TCC Fault Input I/O pin.							

46.27 Universal Serial Bus (USB) Electrical Characteristics

Table 46-39. USB AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDIO 3.0V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics ⁽¹⁾	Min.	Typ.	Max.	Units	Conditions
USB_1	VDDUSB	USB Transceiver Voltage	3	—	3.6	V	Voltage on VDDIOx must be in this range for proper USB operation
Clocks							
USB_23	USBCLKS ⁽²⁾	USB Clock Source	—	48	—	MHz	fGCLK_USBFS = 48 MHz
USB_25	USBAHB	Min. AHB Clock for USB Operations	12	—	—	MHz	
Notes:							
1. Supports Full and Low-Speed communication. Compliant with USB 2.0 specification.							
2. Required clock accuracy: Full-Speed = ±0.25%, Low-Speed = ±1.5%.							

Table 46-40. USB FS/LS AC Electrical Specifications

USB FS/LS Clock configuration			Standard Operating Conditions: VDDIO 3.0V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial	
Param. No.	Symbol	Characteristics	Device Mode Operation	Host Mode Operation
USB_30	USBFS_GCLK_SRC	DFLL48M Open loop	No	No
		DFLL48M Close loop, Ref. internal OSC source	No	No
		DFLL48M Close loop, Ref. external XOSC / XOSC32K source	Yes	No
		DFLL48M Close loop, Ref. SOF (USB recovery mode)	Yes	N/A
		PLL internal OSC (DFLL48M)	No	No
		PLL external OSC	Yes	Yes
		XOSC Crystal	Yes	Yes

46.28 Flash Controller Electrical Characteristics

Table 46-41. Flash NVM AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
NVM_1	FRETEN	Flash Data Retention	20	—	—	Yrs	Under all conditions less than Absolute Maximum Ratings specifications
NVM_3	EP ⁽²⁾	Cell Endurance (Flash Erase and Write Operation) ⁽²⁾	10k	—	—	Cycles	

Table 46-41. Flash NVM AC Electrical Specifications (continued)

AC CHARACTERISTICS				Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics		Min.	Typ.	Max.	Units	Conditions
NVM_5	FREAD ⁽¹⁾	Flash Read ⁽¹⁾	0 Wait States	—	—	(*)	MHz	VDDIOx = 3.3V, FCR.CTRLA.AUTOWS = 0
			1 Wait States	—	—	(*)		
			2 Wait States	—	—	(*)		
			0 Wait States	—	—	(*)	MHz	VDDIOx = 1.8V or VDDIO(min) whichever is greater, FCR.CTRLA.AUTOWS = 0
			1 Wait States	—	—	(*)		
			2 Wait States	—	—	(*)		
NVM_7	TFW	Program Cycle Time	Write Double Word	—	—	(*)	μs	Under all conditions less than Absolute Maximum Ratings specifications
NVM_8	TFPP		Pre-Program Double Word	—	—	(*)	μs	
NVM_9	TCE		Erase Chip	—	—	(*)	ms	
NVM_11	TFEP		Erase Page	—	—	(*)	ms	
NVM_13	IDDPROG	Supply Current during Programming		—	—	PAI_401	mA	VDDIOx = 3.3V

Notes:

- Maximum FLASH operating frequencies are given in the table above, but are limited by the Embedded Flash access time when the processor is fetching code out of it. These tables provide the device maximum operating frequency defined by the field FWS of the FCW CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.
- Cell Endurance is reached with Flash Preprogramming option disabled. Enabling such option will improve the Cell Endurance but slow-down the programming time.

46.29 Frequency Meter (FREQM) Electrical Characteristics

Table 46-42. Frequency Meter AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
FM_1	FMLOW	GCLK_IOx Input Low Time	FMPERIOD/2	—	—	ns	VDDIOx (min) and meet FM5 spec
FM_3	FMHIGH	GCLK_IOx Input High Time	FMPERIOD/2	—	—	ns	
FM_5	FMPERIOD	GCLK_IOx Input Period	1/FCLK_45	—	—	ns	VDDIOx(min)
FM_7	fGCLK_FREQM_REF	FREQM Reference	—	—	FCLK_17	MHz	
FM_9	fGCLK_FREQM_MSR	FREQM Measure	—	—	FCLK_15	MHz	

46.30 JTAG Electrical Characteristics

Figure 46-20. JTAG AC Timing Diagrams

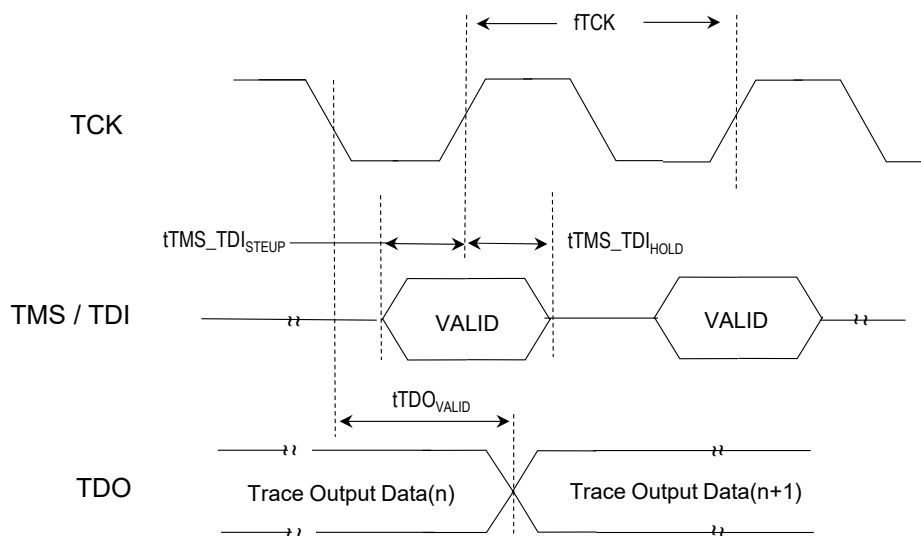


Table 46-43. JTAG AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
JTAG_1	f_{TCK}	JTAG TCK Frequency	—	—	24	MHz	VDDIO(min) - VDDIO(max)
JTAG_3	$t_{TMS\_TDISETUP}$	TMS/TDI Setup Time	2.3	—	—	ns	
JTAG_5	$t_{TMS\_TDIHOLD}$	TMS.TDI Hold Time	0	—	—	ns	
JTAG_7	$t_{TDOVALID}$	TDO Output Valid	—	—	20.6	ns	

46.31 SWD 2-Wire Electrical Characteristics

Figure 46-21. SWD 2-Wire Read/Write AC Timing Diagrams

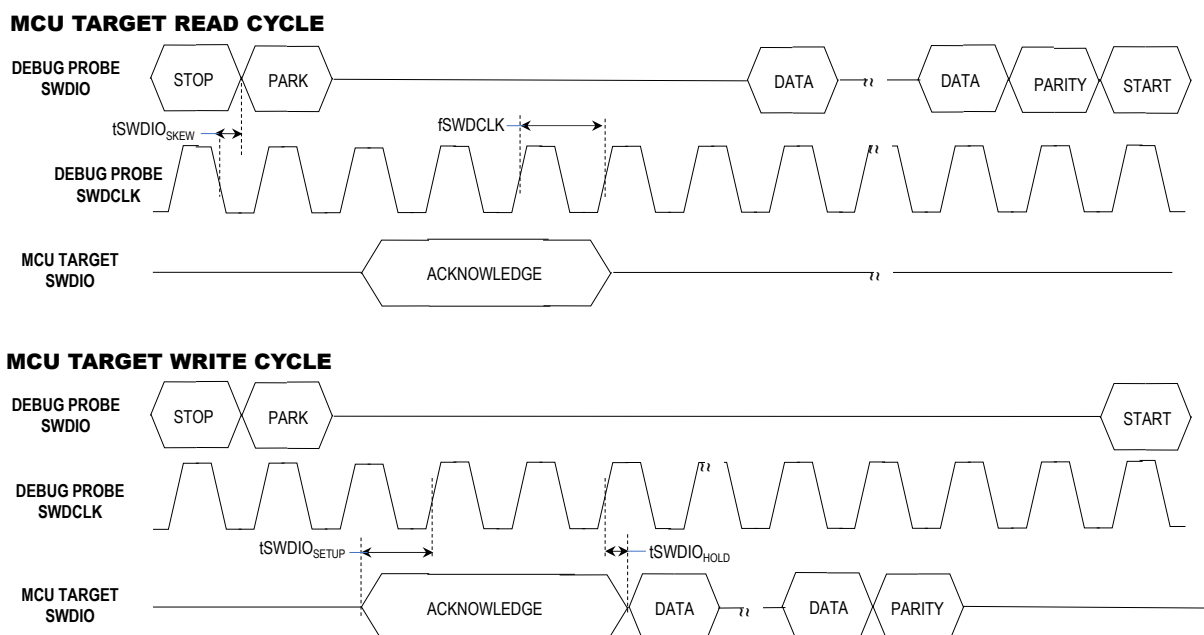


Table 46-44. SWD 2-Wire AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ.	Max.	Units	Conditions
SWD_1	fSWDCLK	SWDCLK Clock Frequency	—	—	40	MHz	VDDIO(min) - VDDIO(max)
SWD_3	tSWDCLKHIGH	SWDCLK Clock High Time	1 / (2 * fSWDCLK)	—	—	ns	
SWD_5	tSWDCLKLOW	SWDCLK Clock Low Time	1 / (2 * fSWDCLK)	—	—	ns	
SWD_7	tSWDIOSKEW	SWDIO Skew	-5	—	5	ns	
SWD_9	tSWDIOSETUP	SWDIO Setup Time	4	—	—	ns	
SWD_11	tSWDIOHOLD	SWDIO Hold Time	1	—	—	ns	

46.32 Configurable Custom Logic (CCL) Electrical Specifications

Table 46-45. CCL AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
TSU_CCL_FF	—	Setup for CCL data to clock for flip-flop modes	2*CCL_GCLK	—	—	ns	—

46.33 User LDO Electrical Specifications

Table 46-46. User LDO AC Electrical Specifications

AC CHARACTERISTICS			Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param. No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
ULDO_1	VIN - VOUT	Dropout Voltage	—	—	0.4	V	IOUT = 100mA
ULDO_3			—	—	0.2	V	IOUT = 30mA
ULDO_5	VOUT	Output Voltage Accuracy	-5	±2	5	%	IOUT = 100µA, ULDOLEVEL 0 (Vout 1.2V)
			-5	±2	5	%	IOUT = 100µA, ULDOLEVEL 1 (Vout 1.5V)
			-5	±2	5	%	IOUT = 100µA, ULDOLEVEL 2 (Vout 1.8V)
			-5	±2	5	%	IOUT = 100µA, ULDOLEVEL 3 (Vout 2.5V)
ULDO_7	ΔVOUT/VOUT	Load Regulation	—	4.6	6.5	%	0mA ≤ IOUT ≤ 100 Ma, ULDOLEVEL 0 (Vout 1.2V)
			—	3.7	5	%	0mA ≤ IOUT ≤ 100 mA, ULDOLEVEL 1 (Vout 1.5V)
			—	3	4.1	%	0mA ≤ IOUT ≤ 100 mA, ULDOLEVEL 2 (Vout 1.8V)
			—	2.2	3	%	0mA ≤ IOUT ≤ 100 mA, ULDOLEVEL 3 (Vout 2.5V)
ULDO_9	IOUT	Output current	0	—	100	mA	—
ULDO_11	ICL	Current Limit	110	120	141	mA	VOUT = 0V

46.34 Multi-Channel Ram Controller (MCRAMC) Electrical Specifications

Table 46-47. MCRAMC Wait State Electrical Characteristics

DC CHARACTERISTICS			Standard Operating Conditions: VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristics	Min.	Typ	Max.	Units	Conditions
MCRAMC_1	FCLK3 ⁽¹⁾	MCRAMC maximum frequency	—	—	0 to 72	MHz	MCRAMC.CTRLA[WS] = 0 No RAM wait state introduced, maximum throughput operation
Note: 1. Programmed via the MCRAMC.CTRLA[WS].							

47. Packaging Information

47.1 Package Marking Information

All devices are marked with the Microchip logo, a shortened ordering code and additional marking (the two last lines)

YYWW R ARM

XXXXXX CC

Where:

- "Y" or "YY": Manufacturing Year (last OR two last digit(s))
- "WW": Manufacturing Week
- "R": Revision
- "XXXXXX": Lot number
- "CC": Internal Code

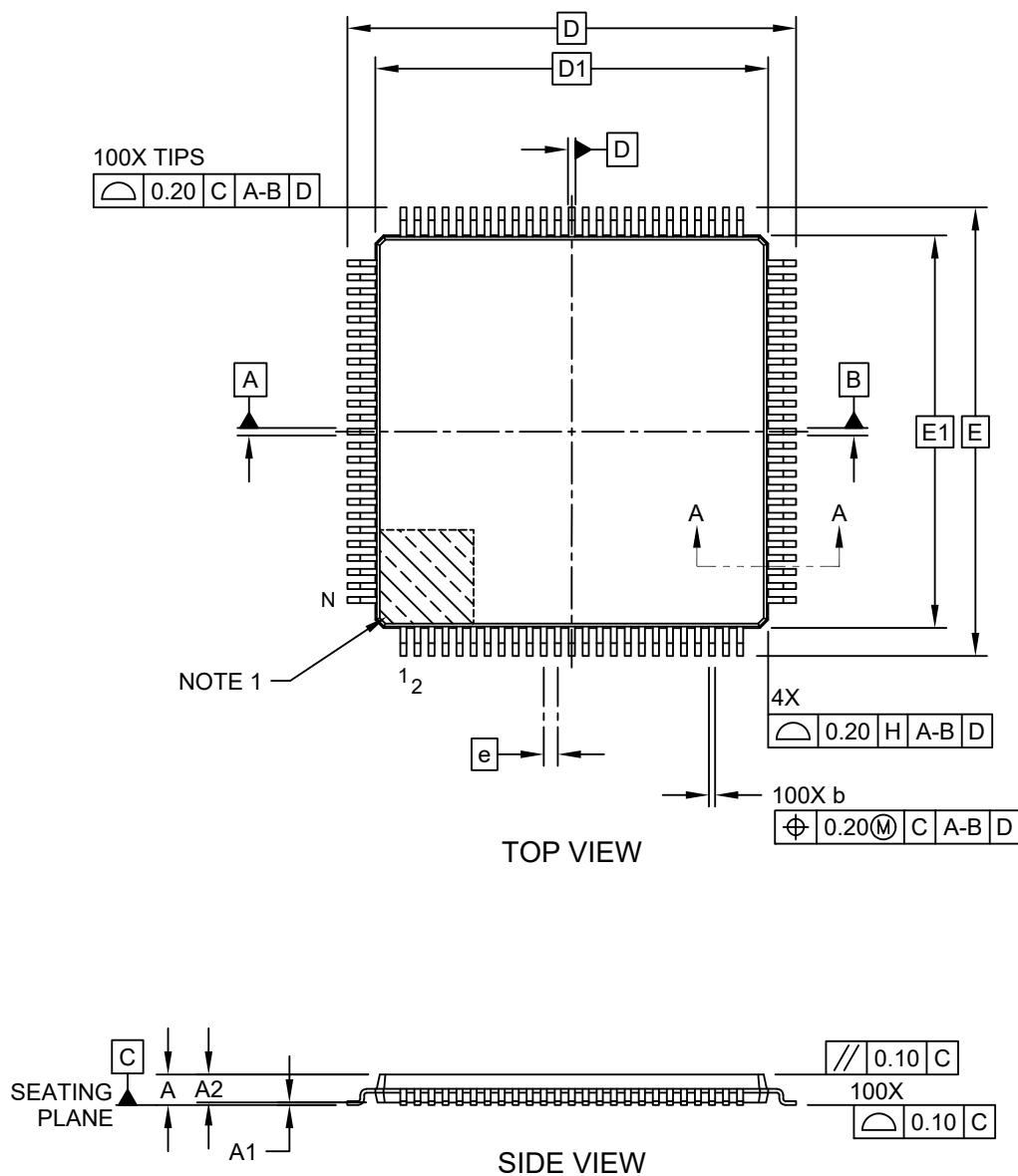
47.2 Package Drawings

Note: For current package drawings, refer to the Microchip Packaging Specification, which is available at www.microchip.com/packaging.

47.2.1 100-pin TQFP

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body [TQFP] 2.00 mm Footprint

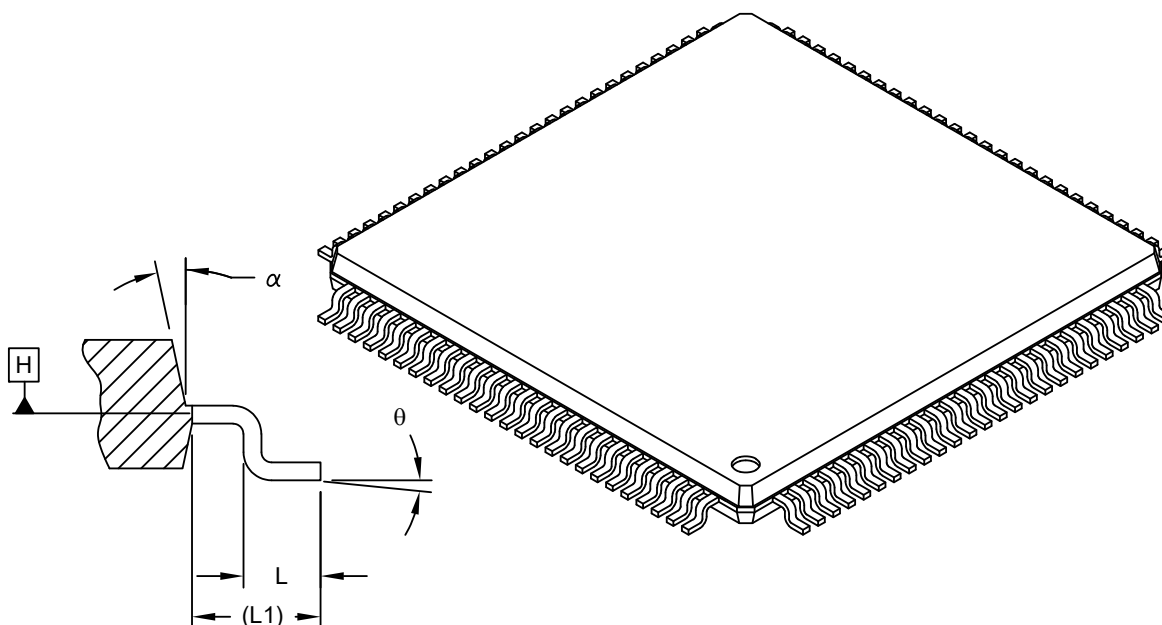
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing C04-110-PF Rev C Sheet 1 of 2

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body [TQFP] 2.00 mm Footprint

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



SECTION A-A

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N	100		
Lead Pitch	e	0.50 BSC		
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	θ	0°	-	7°
Overall Width	E	16.00 BSC		
Overall Length	D	16.00 BSC		
Molded Package Width	E1	14.00 BSC		
Molded Package Length	D1	14.00 BSC		
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	-	13°

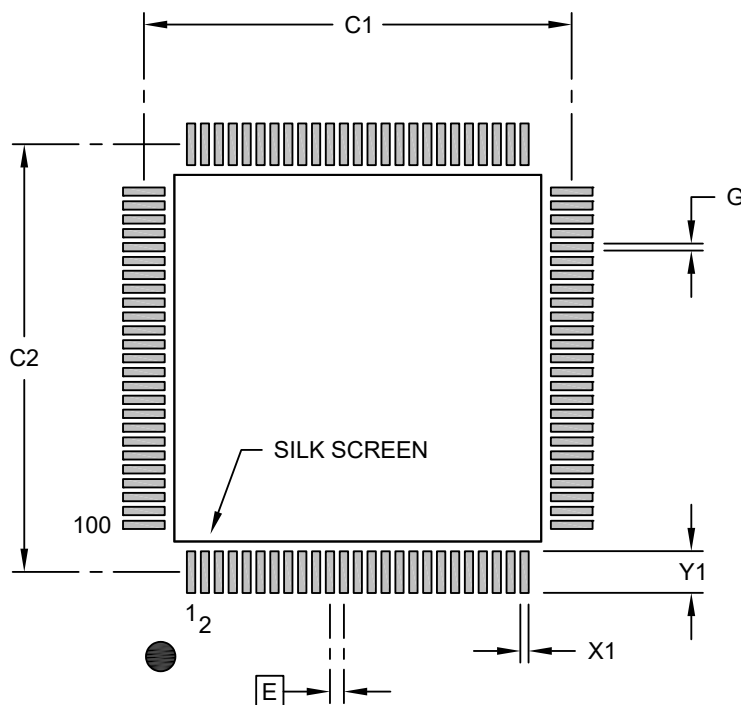
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-110-PF Rev C Sheet 2 of 2

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body [TQFP] 2.00 mm Footprint

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (X100)	Y1			1.50
Contact Pad to Contact Pad (X96)	G	0.20		

Notes:

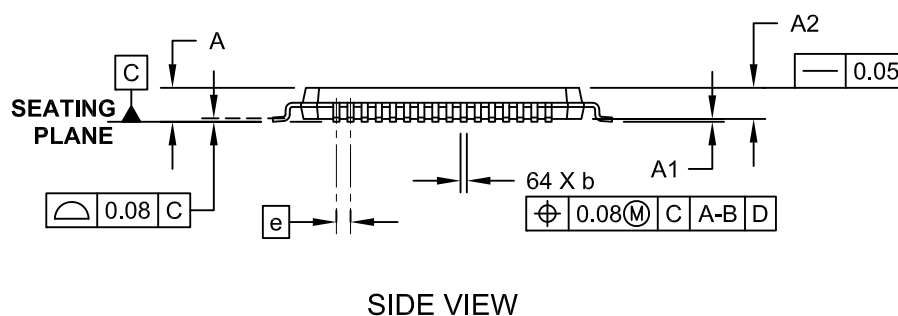
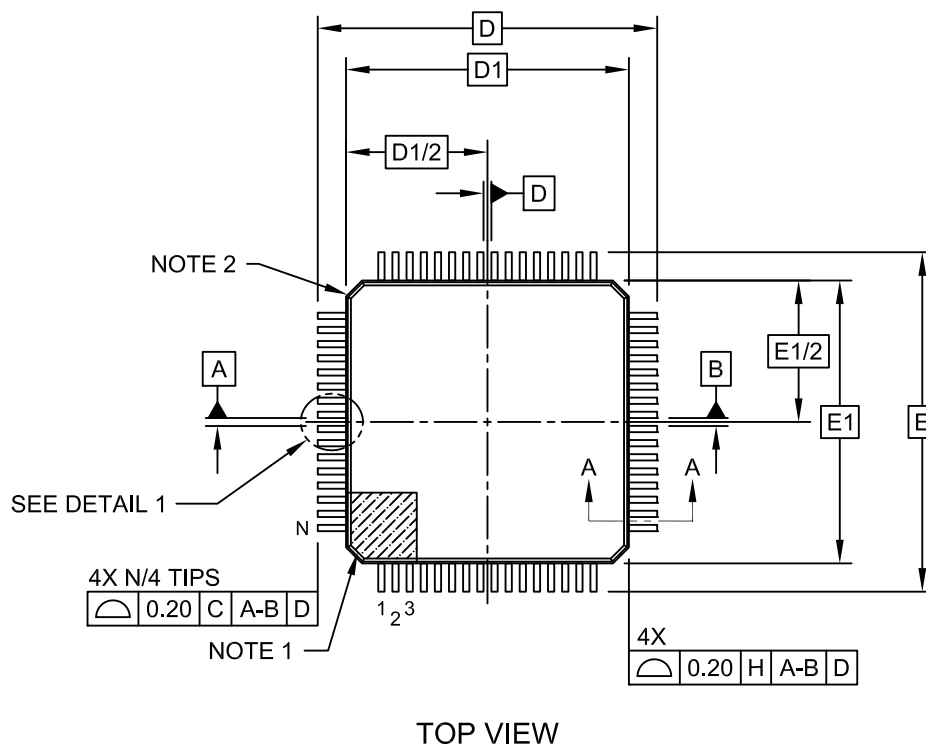
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2110-PF Rev C

47.2.2 64-pin TQFP

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

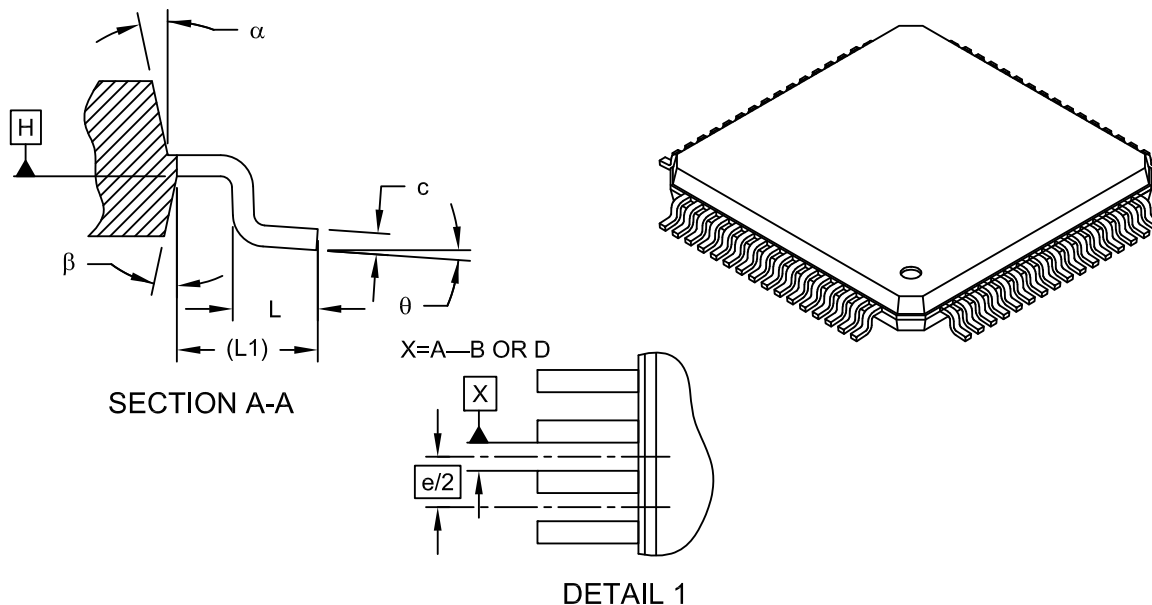
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Eraving C04-085-PT Rev E Sheet 1 of 2

64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



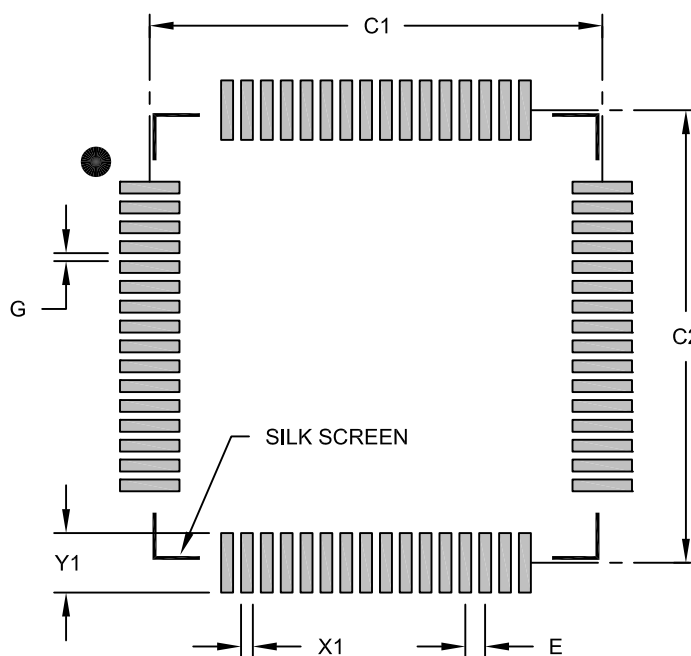
		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Leads	N		64		
Lead Pitch	e		0.50 BSC		
Overall Height	A		-	-	1.20
Molded Package Thickness	A2		0.95	1.00	1.05
Standoff	A1		0.05	-	0.15
Foot Length	L		0.45	0.60	0.75
Footprint	L1		1.00 REF		
Foot Angle	θ		0°	3.5°	7°
Overall Width	E		12.00 BSC		
Overall Length	D		12.00 BSC		
Molded Package Width	E1		10.00 BSC		
Molded Package Length	D1		10.00 BSC		
Lead Thickness	c		0.09	-	0.20
Lead Width	b		0.17	0.22	0.27
Mold Draft Angle Top	α		11°	12°	13°
Mold Draft Angle Bottom	β		11°	12°	13°

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085-PT Rev E Sheet 2 of 2

64-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 2.00 mm Footprint [TQFP]



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

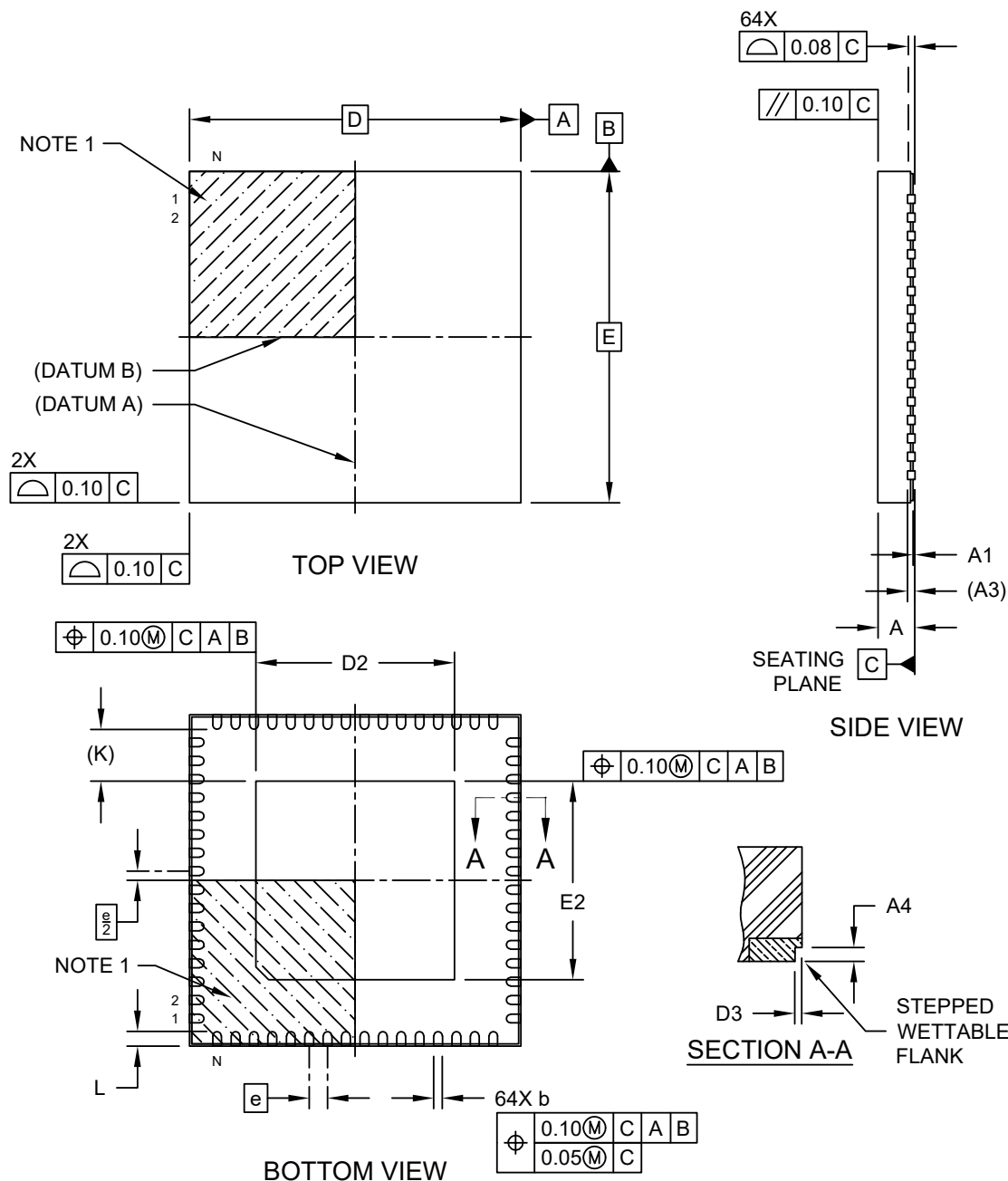
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2085-PT Rev E

47.2.3 64-pin VQFN

64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks

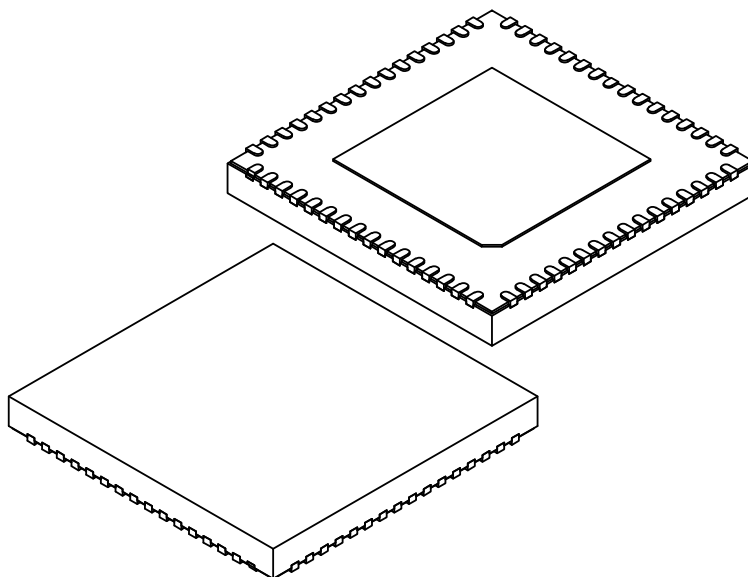
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing C04-483 Rev E Sheet 1 of 2

64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Terminals	N		64		
Pitch	e		0.50 BSC		
Overall Height	A		0.80	0.90	1.00
Standoff	A1		0.00	0.02	0.05
Terminal Thickness	A3		0.203 REF		
Overall Length	D		9.00 BSC		
Exposed Pad Length	D2		5.30	5.40	5.50
Overall Width	E		9.00 BSC		
Exposed Pad Width	E2		5.30	5.40	5.50
Terminal Width	b		0.20	0.25	0.30
Terminal Length	L		0.30	0.40	0.50
Terminal-to-Exposed-Pad	K		1.40 REF		
Wettable Flank Step Length	D3		0.035	0.060	0.085
Wettable Flank Step Height	A4		0.10	-	0.19

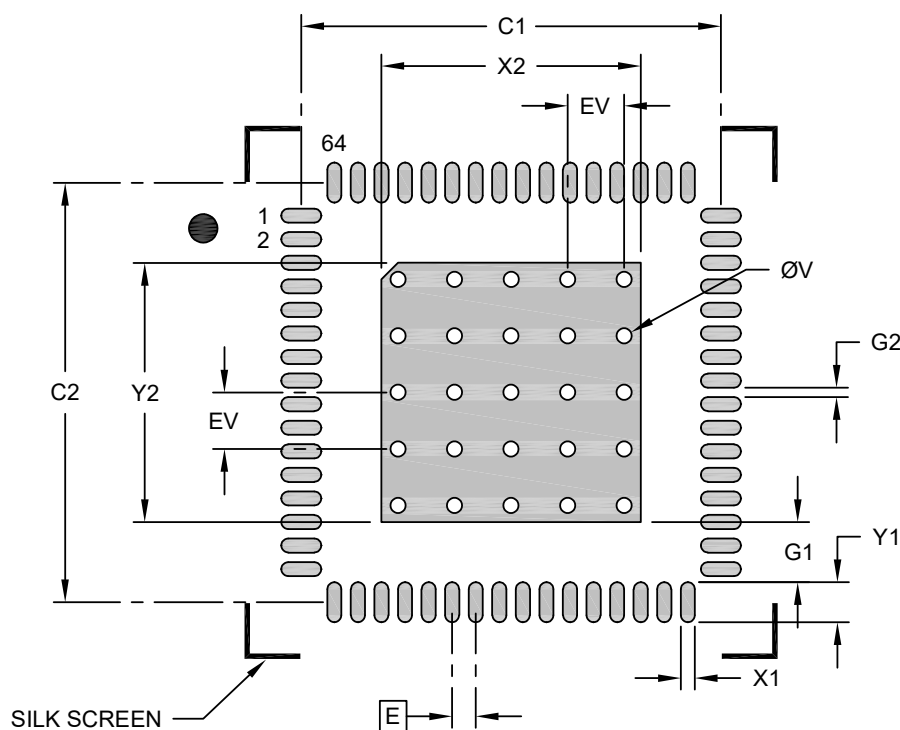
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated
- Dimensioning and tolerancing per ASME Y14.5M
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-483 Rev E Sheet 2 of 2

64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	X2			5.50
Optional Center Pad Length	Y2			5.50
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Contact Pad to Center Pad (X64)	G1	1.28		
Contact Pad to Contact Pad (X60)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

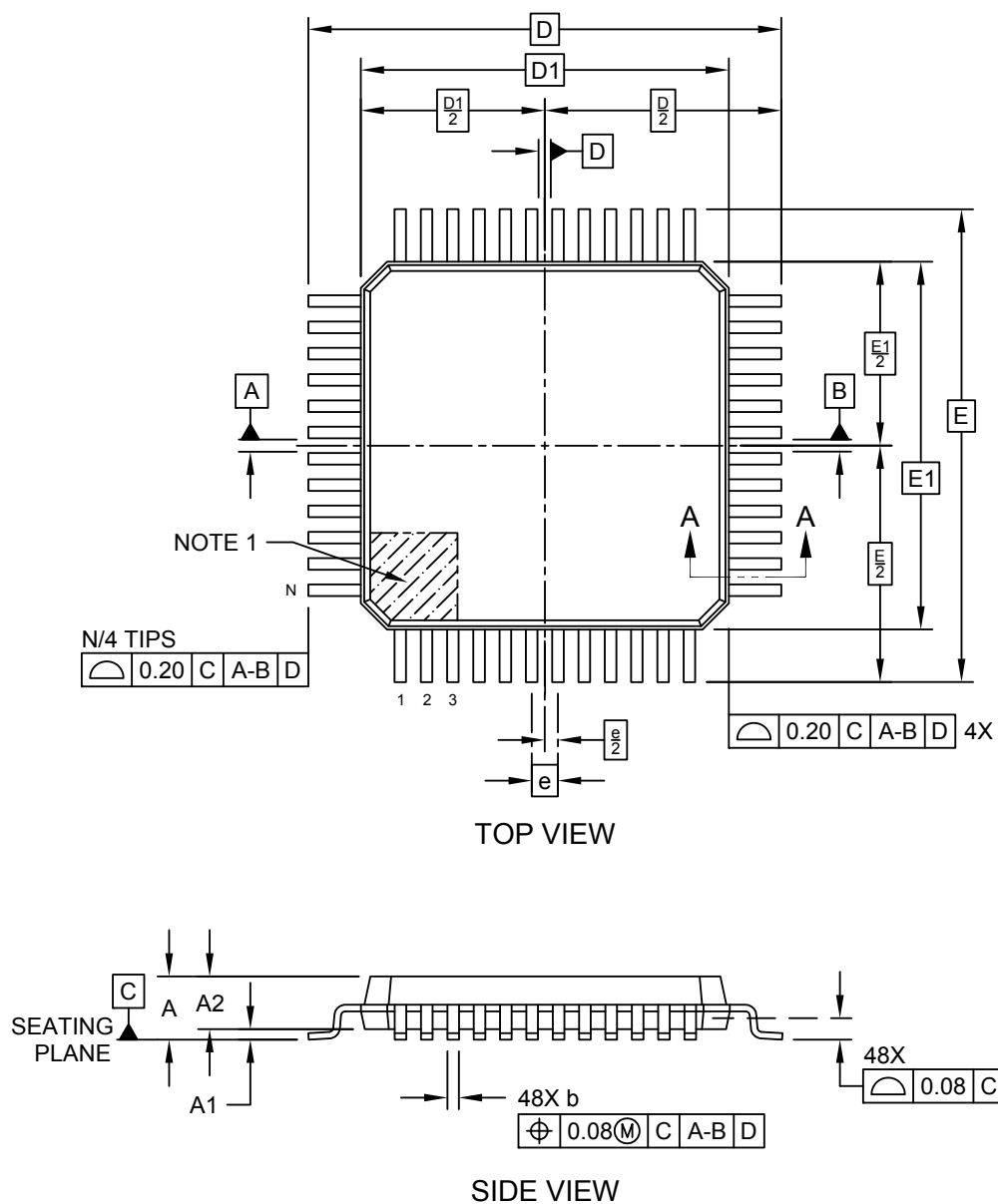
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2483 Rev E

47.2.4 48-pin TQFP

48-Lead Plastic Thin Quad Flatpack (Y8) - 7x7x1.0 mm Body [TQFP]

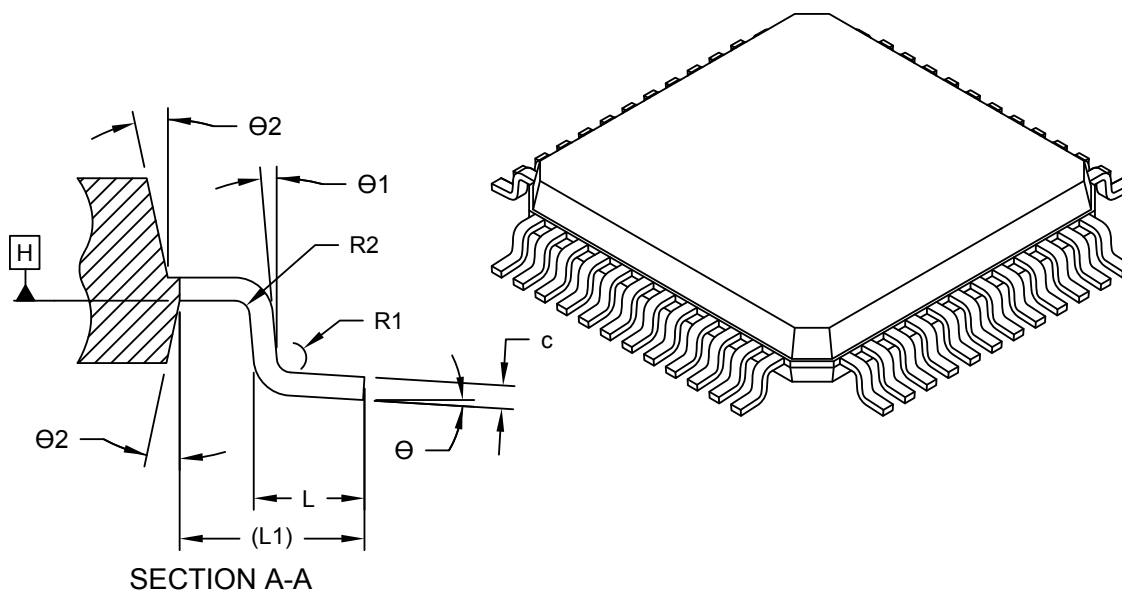
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Microchip Technology Drawing C04-300-Y8 Rev D Sheet 1 of 2

48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Terminals	N	48		
Pitch	e	0.50 BSC		
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Length	D	9.00 BSC		
Molded Package Length	D1	7.00 BSC		
Overall Width	E	9.00 BSC		
Molded Package Width	E1	7.00 BSC		
Terminal Width	b	0.17	0.22	0.27
Terminal Thickness	c	0.09	-	0.16
Terminal Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Lead Bend Radius	R1	0.08	-	-
Lead Bend Radius	R2	0.08	-	0.20
Foot Angle	Θ	0°	3.5°	7°
Lead Angle	Θ1	0°	-	-
Mold Draft Angle	Θ2	11°	12°	13°

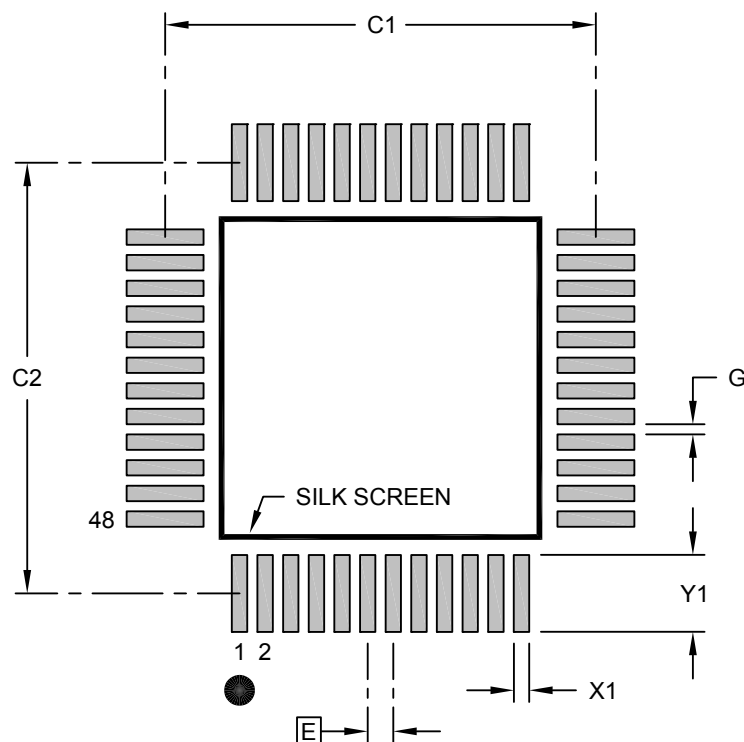
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-300-Y8X Rev D Sheet 2 of 2

48-Lead Plastic Thin Quad Flatpack (Y8) - 7x7x1.0 mm Body [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		8.40	
Contact Pad Spacing	C2		8.40	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

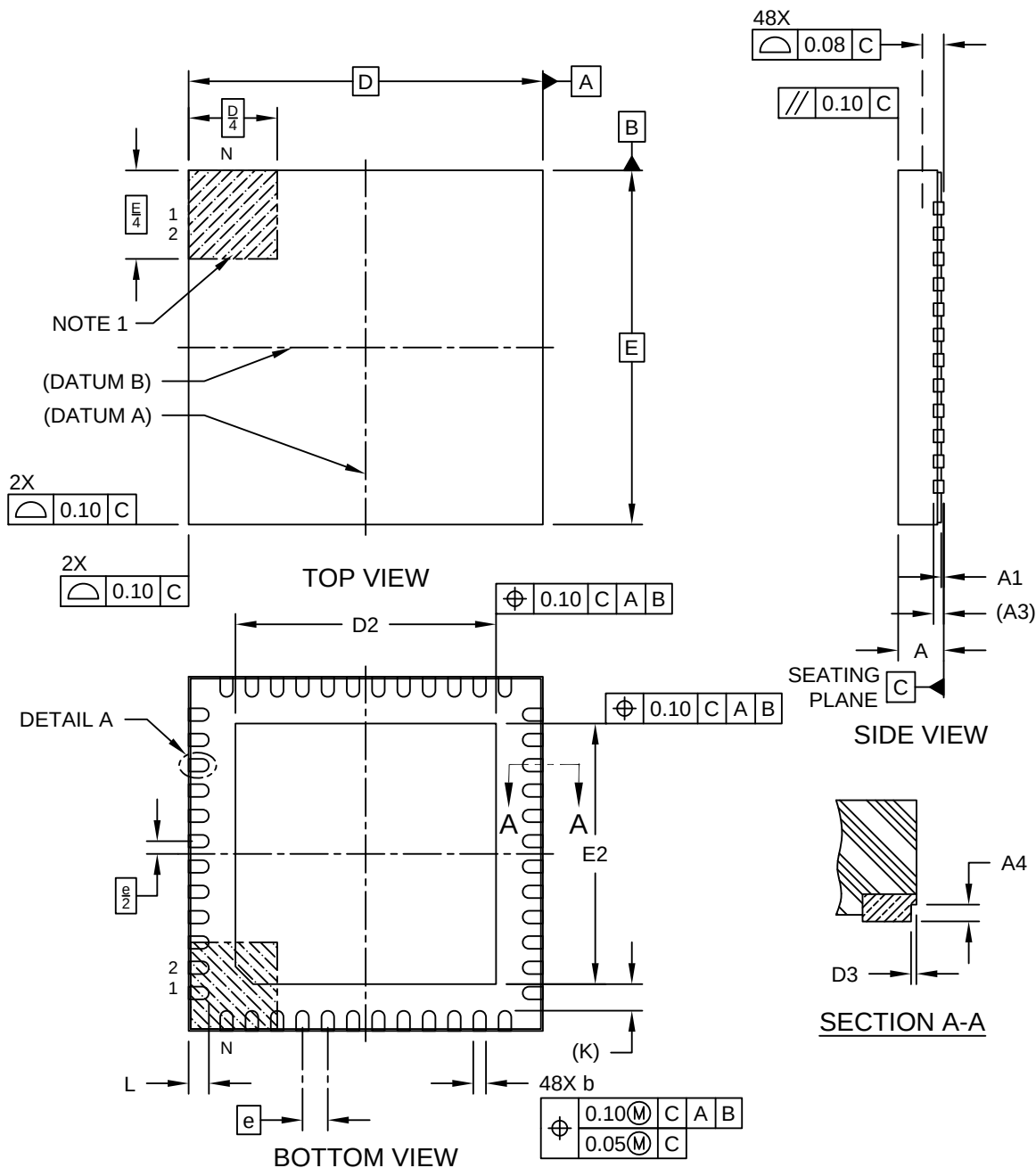
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2300-Y8 Rev D

47.2.5 48-pin VQFN

48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

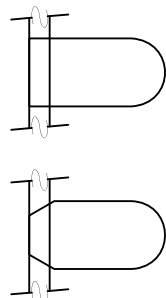
Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



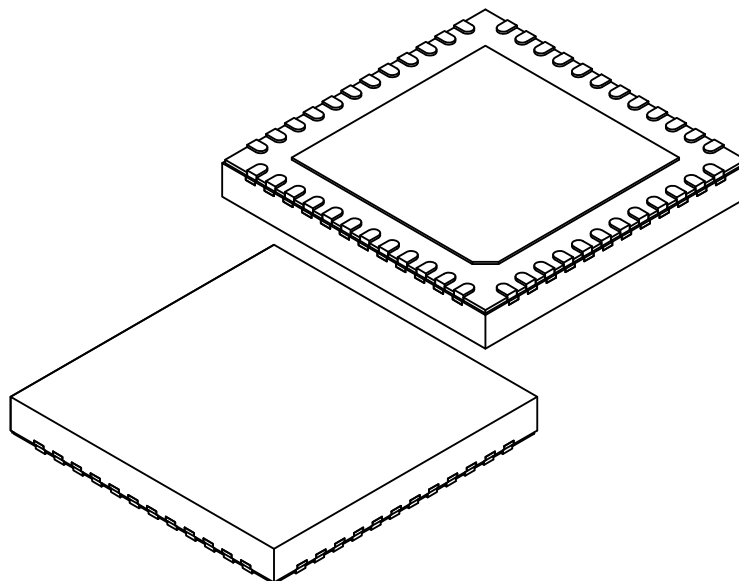
Microchip Technology Drawing C04-21493 Rev A Sheet 1 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



DETAIL 1
ALTERNATE TERMINAL
CONFIGURATIONS



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N	48		
Pitch	e	0.50 BSC		
Overall Height	A	0.80	0.85	0.90
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.203 REF		
Overall Length	D	7.00 BSC		
Exposed Pad Length	D2	5.05	5.15	5.25
Overall Width	E	7.00 BSC		
Exposed Pad Width	E2	5.05	5.15	5.25
Terminal Width	b	0.20	0.25	0.30
Terminal Length	L	0.35	0.40	0.45
Terminal-to-Exposed-Pad	K	0.53 REF		
Wettable Flank Step Length	D3	-	-	0.085
Wettable Flank Step Height	A4	0.10	-	0.19

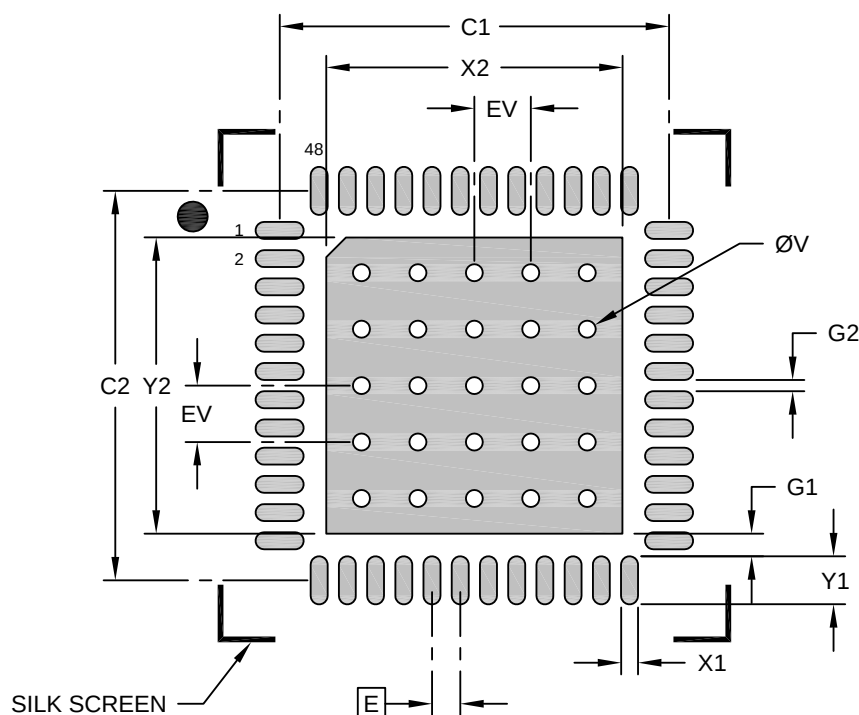
Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated
- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21493 Rev A Sheet 2 of 2

48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Optional Center Pad Width	X2			5.25
Optional Center Pad Length	Y2			5.25
Contact Pad Spacing	C1		6.90	
Contact Pad Spacing	C2		6.90	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	Y1			0.85
Contact Pad to Center Pad (X48)	G1	0.20		
Contact Pad to Center Pad (X44)	G2	0.40		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

Notes:

- Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23493 Rev A

47.3 Soldering Profile

The following table gives the recommended soldering profile from J-STD-20.

Table 47-1. Recommended Soldering Profile

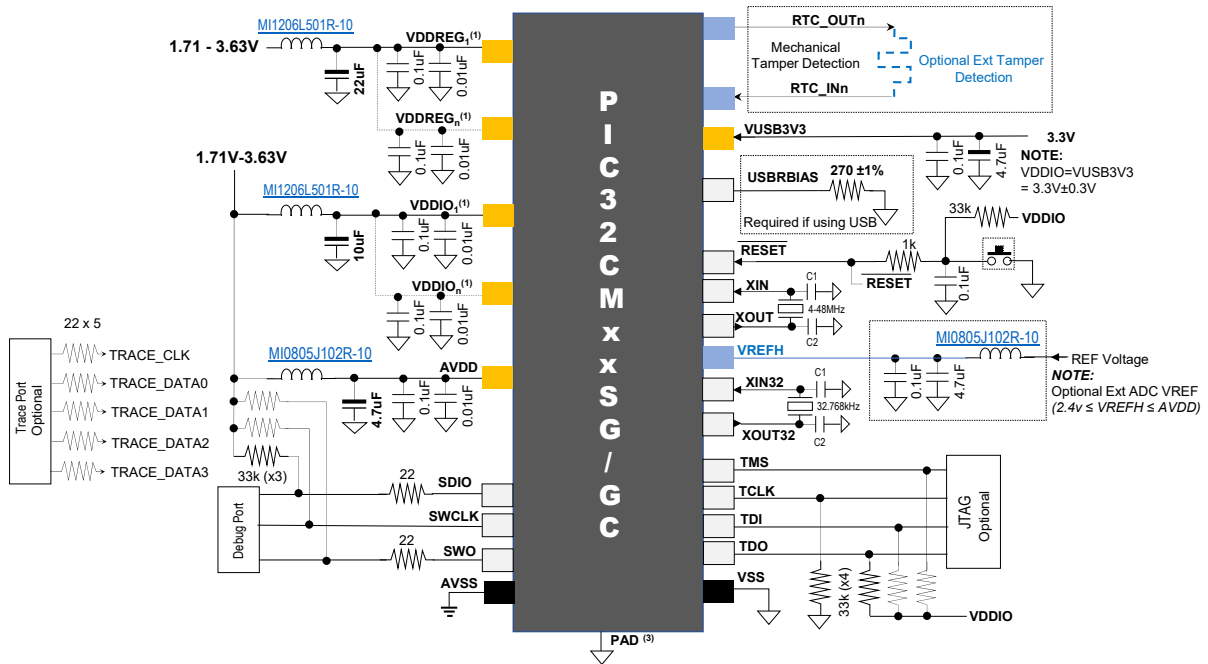
Profile Feature	Green Package
Average Ramp-up Rate (217°C to peak)	3°C/s max.
Preheat Temperature 175°C ±25°C	150-200°C
Time Maintained Above 217°C	60-150s
Time within 5°C of Actual Peak Temperature	30s
Peak Temperature Range	260°C
Ramp-down Rate	6°C/s max.
Time 25°C to Peak Temperature	8 minutes max.

A maximum of three reflow passes is allowed per component.

48. Schematic Checklist

48.1 Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a PIC32CM SG00/GC00 design. This chapter illustrates the recommended power supply connections, how to connect the external analog references, programmer, and debugger.



49. Common Conventions

49.1 Numerical Notation

Table 49-1. Numerical Notation

Symbol	Description
165	Decimal number
0b0101	Binary number (example 0b0101 = 5 decimal)
'0101'	Binary numbers are given without prefix if unambiguous
0x3B24	Hexadecimal number
X	Represents an unknown or do not care value
Z	Represents a high-impedance (floating) state for either a signal or a bus

49.2 Memory Size and Type

Table 49-2. Memory Size and Bit Rate

Symbol	Description
KB (kbyte)	kilobyte ($2^{10} = 1024$)
MB (Mbyte)	megabyte ($2^{20} = 1024 \times 1024$)
GB (Gbyte)	gigabyte ($2^{30} = 1024 \times 1024 \times 1024$)
b	bit (binary '0' or '1')
B	byte (8 bits)
1kbit/s	1,000 bit/s rate (not 1,024 bit/s)
1Mbit/s	1,000,000 bit/s rate
1Gbit/s	1,000,000,000 bit/s rate
word	32 bit
half-word	16 bit

49.3 Frequency and Time

Table 49-3. Frequency and Time

Symbol	Description
kHz	1 kHz = 10^3 Hz = 1,000 Hz
KHz	1 KHz = 1,024 Hz, 32 KHz = 32,768 Hz
MHz	1 MHz = 10^6 Hz = 1,000,000 Hz
GHz	1 GHz = 10^9 Hz = 1,000,000,000 Hz
s	second
ms	millisecond
μs	microsecond
ns	nanosecond

49.4 Registers and Bits

Table 49-4. Register and Bit Mnemonics

Symbol	Description
R/W	Read/Write accessible register bit. The user can read from and write to this bit.
R	Read-only accessible register bit. The user can only read this bit. Writes will be ignored.

Table 49-4. Register and Bit Mnemonics (continued)

Symbol	Description
W	Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value.
BIT	Bit names are shown in uppercase. (Example ENABLE)
FIELD[n:m]	A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0})
Reserved	Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read. Reserved bit field values must not be written to a bit field. A reserved value will not be read from a read-only bit field. Do not write any value to reserved bits of a fuse.
PERIPHERAL/	If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance.
Reset	Value of a register after a Power-on Reset. This is also the value of registers in a peripheral after performing a software Reset of the peripheral, except for the Debug Control registers.
SET/CLR	Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a '1' to a bit in the CLR register will clear the corresponding bit in both registers, while writing a '1' to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence.

50. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

Table 50-1. Acronyms and Abbreviations

Abbreviation	Description
AC	Analog Comparator
ADC	Analog-to-Digital Converter
ADDR	Address
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	AMBA Advanced Peripheral Bus
AREF	Analog Reference Voltage
BOD	Brown-out Detector
CAL	Calibration
CC	Compare/Capture
CCL	Configurable Custom Logic
CLK	Clock
CRC	Cyclic Redundancy Check
CTRL	Control
DAC	Digital-to-Analog Converter
DAP	Debug Access Port
DFLL	Digital Frequency Locked Loop
DMAC	DMA (Direct Memory Access) Controller
DSU	Device Service Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EIC	External Interrupt Controller
EVSYS	Event System
FREQM	Frequency Meter
GCLK	Generic Clock Controller
GND	Ground
GPIO	General Purpose Input/Output
HSM	Hardware Security Module
I ² C	Inter-Integrated Circuit
IF	Interrupt Flag
INT	Interrupt
MBIST	Memory Built-In Self-Test
MEM-AP	Memory Access Port
MTB	Micro Trace Buffer
NMI	Non-maskable Interrupt
NVIC	Nested Vector Interrupt Controller
NVM	Nonvolatile Memory
NVMCTRL	Nonvolatile Memory Controller
OSC	Oscillator
PAC	Peripheral Access Controller
PC	Program Counter

Table 50-1. Acronyms and Abbreviations (continued)

Abbreviation	Description
PER	Period
PLL	Phase Locked Loop
PM	Power Manager
POR	Power-on Reset
PORT	I/O Pin Controller
PUF	Physically Unccloneable Function
PWM	Pulse-Width Modulation
RAM	Random-Access Memory
REF	Reference
RTC	Real-Time Counter
RX	Receiver/Receive
SEEP	SmartEEPROM Page
SERCOM	Serial Communication Interface
SMBus	System Management Bus
SP	Stack Pointer
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SUPC	Supply Controller
SWD	Serial Wire Debug
TC	Timer/Counter
TRNG	True Random Number Generator
TX	Transmitter/Transmit
ULP	Ultra Low-Power
USART	Universal Synchronous and Asynchronous Serial Receiver and Transmitter
V _{DD}	Common voltage to be applied to VDDIO and AVDD
V _{DDIN}	Digital Supply Voltage
V _{DDIO}	Digital Supply Voltage
AV _{DD}	Analog Supply Voltage
VREF	Voltage Reference
WDT	Watchdog Timer
XOSC	Crystal Oscillator

51. Revision History

Revision C - 02/2025

The following updates were performed for this revision:

Section	Updates
General	Multiple formatting updates throughout the document for public release
Product Mapping	All new content
Processor and Architecture	All new content
All Peripherals	- Updated all Registers for each Peripheral with new content - Updated all Peripheral Dependencies Tables with new content - Reformatted the Register Summaries for the following chapters: CAN DMAC USB

Revision History (continued)

Section	Updates
Electrical Characteristics	- Updated Absolute Maximum Electrical Characteristics with new Voltages - Updated CPU Electrical Characteristics with new typ specifications - Updated the following sections with new min and max specifications: Power Supply Active Power Idle Power Standby Power Hibernate Power Backup Power OFF Power Wake-up Timing Peripheral Active Power I/O Pin Electrical Characteristics Internal Voltage Reference Electrical Specifications (IVREF) Maximum Clock Frequencies XOSC Electrical Characteristics XOSC32K Electrical Characteristics OSCULP32K Electrical Characteristics DFLL-FDPLL Electrical Characteristics Analog-to-Digital Converter (ADC) Electrical Specifications Comparator (AC) Electrical Characteristics Peripheral Touch Controller (PTC) Electrical Characteristics Serial Peripheral Interface (SPI) Timings Electrical Characteristics Universal Asynchronous Receiver/Transmitter (UART) Electrical Characteristics I²C Electrical Characteristics Controller Area Network (CAN) Electrical Characteristics Timer/Counter Controller (TCC) Electrical Characteristics Universal Serial Bus (USB) Electrical Characteristics Flash Controller Electrical Characteristics Frequency Meter (FREQM) Electrical Characteristics JTAG Electrical Characteristics SWD 2-Wire Electrical Characteristics Configurable Custom Logic (CCL) Electrical Specifications User LDO Electrical Specifications Multi-Channel Ram Controller (MCRAMC) Electrical Specifications

Revision B - 06/2024

The following updates were performed for this revision:

Section	Updates
Power Domain Overview	Replaced the Block Diagram with new content
PIC32CM SG00/GC00 Security Features	- Removed erroneous HSM content - Removed erroneous <i>Mix-Secure Peripherals</i> content
NVMCTRL - FCW	Updated the LBWP bitfield of the LBWP Register with a new bitfield number designation
NVMCTRL - FCR	Removed erroneous DERR information from Interrupts
Controller Area Network (CAN)	- Updated Message RAM with a new bitfield designation - Added a new base Address value to the OFFSET bitfield for the CTRLB Register - Updated the following registers with adjusted bitfield values: - SIDFC - XIDFC - RXF0C - RXBC - RXF1C - TXBC - TXEFC - Renamed the TCBS bitfield to TBCS in the TSCFG Register
Hardware Security Module Lite (HSM-Lite)	Updated product naming and added new content
Physically Unccloneable Function (PUF)	Removed obsolete content

Revision A - 02/2024

This is the initial released version of this document.

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>.

ISBN: 979-8-3371-0687-8

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Product Page Links

[PIC32CM5112GC00048](#), [PIC32CM5112GC00064](#), [PIC32CM5112GC00100](#), [PIC32CM5112SG00048](#),
[PIC32CM5112SG00064](#), [PIC32CM5112SG00100](#)